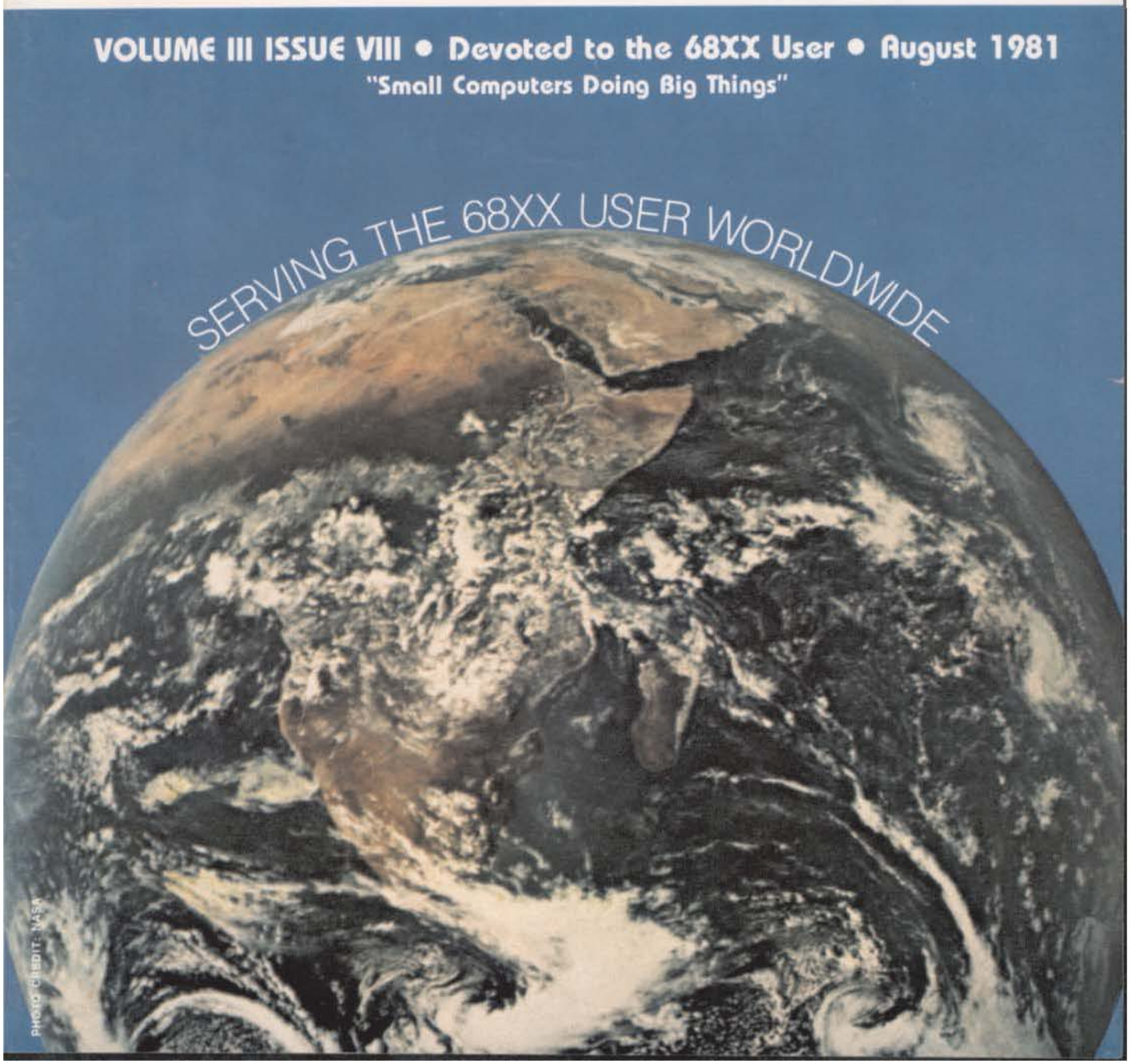
'68' MICRO JOURNAL

$2.50 USA

Australia A $ 4.00 New Zealand NZ $ 4.00
Singapore S $ 8.00 Hong Kong H $20.00
Malaysia M $ 8.00

VOLUME III ISSUE VIII • Devoted to the 68XX User • August 1981

"Small Computers Doing Big Things"



YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or **8209** — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

UniFLEX™

Multi-User

UniFLEX is the first full capability multi-user operating system available for microprocessors. Designed for the 6809 and 68000, it offers its users a very friendly computing environment. After a user 'logs-in' with his user name and password, any of the system programs may be run at will. One user may run the text editor while another runs BASIC and still another runs the C compiler. Each user operates in his own system environment, unaware of other user activity. The total number of users is only restricted by the resources and efficiency of the hardware in use.

Multi-Tasking

UniFLEX is a true multi-tasking operating system. Not only may several users run different programs, but one user may run several programs at a time. For example, a compilation of one file could be initiated while simultaneously making changes to another file using the text editor. New tasks are generated in the system by the 'fork' operation. Tasks may be run in the background or 'locked' in main memory to assist critical response times. Inter-task communication is also supported through the 'pipe' mechanism.

Support

The design of UniFLEX, with its hierarchical file system and device independent I/O, allows the creation of a variety of complex support programs. There is currently a wide variety of software available and under development. Included in this list is a Text Processing System for word processing functions, BASIC interpreter and precompiler for general programming and educational use, native C and Pascal compilers for more advanced programming, sort/merge for business applications, and a variety of debug packages. The standard system includes a text editor, assembler, and about forty utility programs. UniFLEX for 6809 is sold with a single CPU license and one years maintenance for $450.00. Additional yearly maintenance is available for $100.00. OEM licenses are also available.

FLEX™

UniFLEX is offered for the advanced microprocessor systems. FLEX, the industry standard for 6800 and 6809 systems, is offered for smaller, single user systems. A full line of FLEX support software and OEM licenses are also available.

technical systems consultants, inc.
Box 2570, West Lafayette, IN 47906
(317) 463-2502 Telex 276143

'68' MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-CDS1-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

Publisher: Don Williams Sr.

Executive Editor: Larry Williams

Subscriptions and Office manager
Mary Robertson

General Girl 'Friday'
Joyce Williams

Contributing Editors:

Dr. Chuck Adams
Dr. Theo Elbert
Dr. Jeffery Brownstein
Dale Puckett
Russell Gore
Ron Anderson
John Jordon

CONTENTS

Send All Correspondence To:

'68' Micro Journal
3018 Hamill Rd.
PO Box 849
Hixson, Tennessee 37343
— Phone —
Office: 615-870-1993

'68' Micro Journal is published 12 times a year by '68' Micro Journal, 6131 Airways Blvd., Chattanooga, TN 37421. Second Class postage paid at Chattanooga, TN. Postmaster: Send Form 3579 to '68' Micro Journal, PO Box 849, Hixson, TN 37343.

1-Year $18.50 2-Year $32.50 3-Year $48.50

—ITEMS SUBMITTED FOR PUBLICATION—

(Letters to the Editor for Publication) All 'letters to the Editor' should be substantiated by facts. Opinions should be indicated as such. All letters must be signed. We are interested in receiving letters that will benefit or alert our readers. Praise as well as gripes is always good subject matter. Your name may be withheld upon request. If you have had a good experience with a 6800 vendor please put it in a letter. If the experience was bad put that in a letter also. Remember, if you tell us who they are then it is only fair that your name 'not' be withheld. This means that all letters published, of a critical nature, cannot have a name withheld. We will attempt to publish 'verbatim' letters that are composed using 'good taste.' We reserve the right to define (for '68' Micro) what constitutes 'good taste.'

(Articles and items submitted for publication) Please, always include your full name, address, and telephone number. Date and number all sheets. TYPE them if you can, poorly handwritten copy is sometimes the difference between go, no-go. All items should be on 8X11 inch, white paper. Most all art work will be reproduced photographically, this includes all listings, diagrams and other non-text material. All typewritten copy should be done with a NEW RIBBON. All hand drawn art should be black on white paper. Please no hand written code items over 50 bytes. Neatly typed copy will be directly reproduced. Column width should be 3¼ inches.

(Advertising) Any Classified: Maximum 20 words. All single letters and/or numbers will be considered one (1) word. No Commercial or Business Type Classified advertising. Classified ads will be published in our standard format. Classified ads $7.50 one time run, paid in advance.

Commercial and/or Business advertisers please write or phone for current rate sheet and publication lag time.

GIMIX 2MHZ 6809 PLUS 56K SYSTEM

IDEAL FOR SOFTWARE DEVELOPMENT

- You can order a system to meet your needs or select the one featured below.
- **INCLUDES:** 6809 plus CPU #05, Mainframe Cabinet, Mother Board, Power Supply, Fan, 2 Port Serial Card & Cables, 2 Disk Regulator Cards, and **56K**B Static Ram. . . **$2498.29**
- For 50 Hz Export Power Supply, add **$30.00**
- 6800 CPU'S AND SYSTEMS ALSO AVAILABLE.

DUAL DRIVE 5¼'' Disk Systems For Use In GIMIX Mainframes.

- *All Systems Include:* Disk Controller, Cable, and **GIMIX** version of TSC'S Flex.
- Power for the drives is provided by the C.V. power supply in the **GIMIX** Mainframe and 2 of our disk regulator boards. This gives your disk system the same brownout protection and power supply reliability as the rest of the system.
- When ordered with a **GIMIX** 6809 system, GMXBUG 09 and Boot Prom is also included, or, subject to availability, you may substitute Microware's OS-9 for the GMXBUG/FLEX combination at no charge, or have both, Software Selectable installed on the CPU, for **$150.00 Additional.**
- **Systems using Uniflex or Video based will also be available.**
- **To substitute the GIMIX DMA Controller for the #28, Add $200.00 to the Double Density chart.**

SINGLE DENSITY 2 DRIVE SYSTEM USING OUR #48 DISK CONTROLLER

CAPACITY IN BYTES

WITH	FORMATTED	UNFORMATTED	PRICE
2-40 Track (48TPI) Single Sided	199,680	250,000	$ 998.00
2-40 Track (48TPI) Double Sided	399,360	500,000	1198.00
2-80 Track (96TPI) Single Sided	404,480	500,000	1198.00
2-80 Track (96TPI) Double Sided	808,960	1,000,000	1598.00

DOUBLE DENSITY 2 DRIVE SYSTEMS USING OUR #28 DISK CONTROLLER

CAPACITY IN BYTES

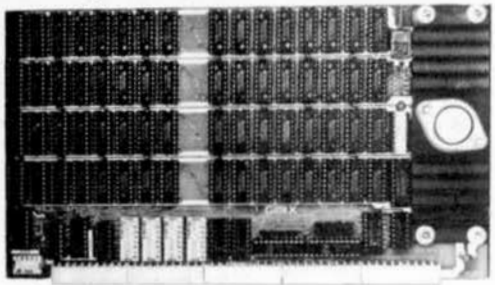
WITH	FORMATTED	UNFORMATTED	PRICE
2-40 Track (48TPI) Single Sided	341,424	500,000	$1148.00
2-40 Track (48TPI) Double Sided	718,848	1,000,000	1348.00
2-80 Track (96TPI) Single Sided	728,064	1,000,000	1348.00
2-80 Track (96TPI) Double Sided	1,456,128	2,000,000	1748.00

THE CLASSY CHASSIS $998.19

32K STATIC RAM BOARDS

Designed for use with:

★ **Existing SS50 Systems** ★ **SS50C Extended Address Systems**



- Assembled
- Burned In
- Tested

16K . .298.12
24K . .348.14
32K . .398.15

16K and 24K Versions are socketed for 32K and require only additional 2114's for expansion

FEATURES:

- Decoding for 4 Extended Address Lines (allows memory decoding up to 1 megabyte)
- DIP-switch to set extended addressing or disable it
- 4 separate 8K blocks, addressable to any 8K boundary by DIP-switch
- Each 8K block may be individually disabled
- Write protect either of two 16K sections
- Low power consumption — uses 2114L low power RAMS
- Fully Socketed
- Gold Bus Connectors
- Guaranteed 2MHz operation

2114 SUPER LOW POWER 200ns RAM CHIPS $2.90

TO ORDER BY MAIL

SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE
Please allow 3 weeks for personal checks to clear

U.S. orders add $5 handling if order is under $200.00. Foreign orders add $10 handling if order is under $200.00

Foreign orders over $200.00 will be shipped via Emery Air Freight COLLECT and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise waiting money. (If checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, account #73 32033. Visa or Master Charge also accepted.

GIMIX INC. reserves the right to change pricing and product specifications at any time without further notice.

GIMIX® and GHOST® are registered trademarks of GIMIX Inc. © 1980 GIMIX Inc.

Flex and Uniflex are trademarks of Technical Systems Consultants Inc. OS-9 is a trademark of Microware Inc. See their ads for other GIMIX compatible software.

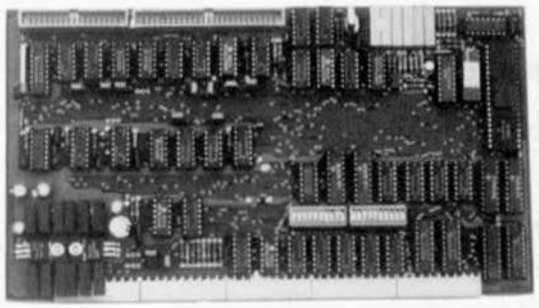
I/O BOARDS

for the 30 PIN BUS:

1 Port Serial (RS 232 or 20MA, current loop)	$ 88.41
2 Port RS 232 Serial	128.43
2 Port Parallel	88.42

for the 50 PIN BUS:

8 Port RS 232 Serial	288.40
8 Port RS 232 Serial with on board Baud Rate generator	318.46
8 Port Parallel	198.45



The GIMIX DMA DISK CONTROLLER ($548.68) uses high speed Direct Memory Access (DMA) for data transfers to and from system memory. It supports any combination of drives, 8" and 5"; single and double headed; single and double track (48 and 96 TPI); single and double density; up to 4 drives total. The board features both a phase-locked loop data separator and adjustable write precompensation to insure high reliability, and can be used in 6809 systems running at 1, 1.5, and 2 MHz.

Available software includes GIMIX versions of the 6809 FLEX disk operating system, $90.00. OS-9 and UniFLEX will also be available.

GIMIX INC.

The Company that delivers
Quality Electronic products since 1975.

1337 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

SEE GHOST AD PAGES 43, 46, 48, & 56

BASIC09™ has a dual personality.

One craves meat-and-potatoes BASIC.

The other prefers Programme ala Pas al.

Some people say BASIC09 is really a PASCAL in disguise, others say it's still BASIC. You'll understand this delightful dilemma when you look at both versions of the "bubble sort" program shown below: both can be run by BASIC09. The program on top is unstructured and hard to understand, but it's traditional BASIC. The program on the bottom is well-structured and easy to follow, a virtue of PASCAL. With BASIC09 you can program either way, or mix the best of both. It's like getting two languages for the price of one.

SORT AN ARRAY
IN ASCENDING SEQUENCE

```
90   DIM A(5)
100  I=5
110  IF I=1 THEN 200
120  FOR J=1 TO I-1
130  IF A(J)<=A(J+1) THEN 170
140  T=A(J+1)
150  A(J+1)=A(J)
160  A(J)=T
170  NEXT J
180  I=I-1
190  GOTO 110
200  RETURN
```

```
DIM array(5)
outer=5
WHILE outer>1 DO
  outer=outer-1
  FOR inner=1 TO outer
    IF array(inner)>=array(inner+1) THEN
      temp=array(inner+1)
      array(inner+1)=array(inner)
      array(inner)=temp
    ENDIF
  NEXT inner
ENDWHILE
RETURN
```

Makes programs better

BASIC09 has five kinds of loop structures: WHILE . . DO, REPEAT . . UNTIL, LOOP . . ENDLOOP, FOR . . NEXT and IF . . THEN . . ELSE. If one of the five built-in data types (byte, integer, real, string, and boolean) doesn't suit the problem, you can make a new one of your liking with the TYPE statement. Need a tree, linked list, or symbol table? Complex non-rectangular data structures using any combination of data types are easy to define. Modular programming breaks down large programs to smaller, more manageable elements. BASIC09 lets you create independent program modules called "procedures" with local variables for recursion plus parameter passing to any other BASIC09 or machine language procedure. There is a complete set of statements for device-independent sequential or random I/O, plus a superlative PRINT USING system.

Makes programs faster

No full-feature BASIC for any 8-bit microprocessor is faster than BASIC09, because it is an interactive compiler. As each program line is entered, it is instantly compiled to a smaller, faster form. Because BASIC09 automatically converts programs back to original "source" form for listing, it is as friendly and easy-to-use as traditional interpreter BASICs. Each procedure can be independently compiled to position-independent, reentrant, ROMable format. Microware® developed a new ultra-fast 9-digit-accuracy floating point math system just for BASIC09. And if that's still not fast enough, there's BYTE and INTEGER arithmetic.

Features that make programs easier to write

The compiler is integrated with a full-feature string AND line-number oriented text editor. If you make a mistake, BASIC09 tells you instantly. String-oriented commands such as search, change, change all occurances, delete, and insert can be used on programs with or without line numbers. There's an automatic line renumbering function too.

Features that make programs easy to test

Debugging often takes longer than writing a program. That's why BASIC09's integral high-level debugger sets it apart from all other compiled OR interpretive languages. The TRACE command shows you each statement executed in BASIC form, plus the result of any expression evaluation. STEP lets you run one or more statements at a time. LET and PRINT allow you to examine or change the values of variables, by name. STATE lists procedure calling order. And there are nine other debug commands. If you need to correct a program, you can edit, recompile, and rerun it in seconds.

Microware® software is available for most popular 6809 computer systems. Source listings and yearly maintenance update service are sold separately for most programs.

Write or call for our free catalog. We accept phone orders and MasterCard and VISA orders.

MICROWARE

Microware Systems Corporation
5835 Grand, Des Moines, Ia 50312
(515) 279-8844 TWX 910-520-2535

Does timesharing on a small system make sense?

It does with OS-9™ Level One!

Now two (or more) acts can share your microcomputer stage. You will no longer have to walk away from your computer while it is busy running a long program. Because OS-9 is a multitasking operating system, you can be running a BASIC program while editing a PASCAL program, for example. This lets you make more efficient use of your time and your system, even if you only use one terminal. If your application requires multiple, independent terminals, one OS-9 system can do the work of several single-user systems.

The convenience of an advanced operating system

Sophistication does not require complexity. Many OS-9 users say that it is actually easier to use than the older 6800-type operating systems. Consider how easy it is to run multiple programs: to run a program you just type its name and hit 'return.' To run a program as a separate job, you type its name, an '&' character, then hit return. The program runs as usual, but OS-9 comes back immediately and is ready for your next command. Simple commands let you see each program's status, set its priority, or abort it.

The file management system has fast, byte-addressable random-and sequential-access files. The tree-structured multiple directory system lets you create separate disk directories for each user, project, or application. Command line I/O file redirection means you specify what device and/or files a program will use when you run it, not when you write it.

Efficiency and hardware versatility

No other operating system can run on such a broad range of hardware: the overall RAM requirement for Level One is 32K to 56K RAM. Memory utilization is superlative because OS-9 lets multiple tasks "share" the same reentrant program. For example, if two users run BASIC09, only one "copy" is actually loaded into memory. The Level Two version of OS-9 can utilize up to a megabyte of memory on systems having memory management hardware (both versions come with complete timesharing support).

OS-9's device independent I/O system can handle almost any number and combination of I/O devices: five or eight inch diskettes, winchester disks, disk cartridges, serial and parallel ports, memory-mapped video displays, and more.

Microware® offers a large selection of "stock" device interface software modules, or you can create your own: all the information you need is in the manuals.

Excellent support and documentation

Each OS-9 package comes with a User's Manual and a System Programmer's Manual that cover every aspect of OS-9. If you have special requirements, you can even purchase the Source Code for most of OS-9 and related software. At Microware® we take pride in offering the best customer support in the business. Technical advice and assistance by phone, mail or telex is available during all business hours.

Superb software tools

In addition to BASIC09, Microware® offers: an Interactive Assembler, Macro Text Editor, Stylograph Word Processor, Interactive Debugger, and coming soon, COBOL, PASCAL and C language compilers.

Microware® software is available for most popular 6809 computer systems. Source listings and yearly maintenance update service are sold separately for most programs.

Please call or write for our free catalog. We accept phone orders and MasterCard and VISA orders.

MICROWARE®

Microware Systems Corporation
5835 Grand, Des Moines, Ia 50312
(515) 279-8844 TWX 910-520-2535

FORTH

FLEX COMPATIBLE FORTH

By Charles (Chuck) Eaker, Ph.D

"This is obviously the most comprehensive manual that's ever been produced on FORTH. It's vastly more complete than anything else! The way he talks about things is not only good reading, but he makes it easy to pick up on the first try."

Said Ron Anderson, '68 MICRO JOURNAL's contributing editor, talking about the X-FORTH manual.

X-FORTH is the best FORTH there is for 6809/6800 computers running FLEX! There's no better way to put it, X-FORTH beats the competition hands down and here's why.

FLEX COMPATIBLE
She can read and write FLEX random and sequential files. She can even read and write the sequential files RANDOMLY! Uses FLEX I/O for terminal and printer. Honors TTYSET.

TWO EDITORS
She has a TTY editor modeled after the FORTH INC. editor rather than the FIG version.
She has a FULL SCREEN EDITOR for terminals that support cursor addressing.

STANDARD ASSEMBLER
She uses standard MOTOROLA mnemonics thus:
'LDA [44]' becomes [44] LDA' in X-FORTH
6809 assembler also supports 6800 mnemonics!

ERROR CONTROL
You the programmer have complete control over disk related errors while other errors provide long error messages. X-FORTH has a protected dictionary and is very hard to accidentally crash!

FANTASTIC DOCUMENTATION (We're very proud of this)
4 Part Ring Bound Manual (over 400 pages!)

1... **GOING FORTH**
Tutorial on the use of FORTH that makes it a snap to learn!!

2... **GOING X-FORTH**
X-FORTH extensions and FLEX interface.
Shows you what the 'X' in X-FORTH is all about!

3... **USER'S MANUAL, DOCUMENTATION AND GLOSSARY**
This is the main part of X-FORTH with sections on: math operations, stacks, input words, output words, strings, disk I/O, edit, editor, assembler, utilities, etc. ALL grouped by function for ease of use! Glossary listing included with each section.

4.. **GLOSSARY AND SOURCE LISTING**
Glossary sorted alphabetically in addition to that in the users manual. Source of all but the core of X-FORTH (also included on disk!)

String primitives, Complete DATA FILE VOCABULARY, Triple precision math package, Portability between 6800 and 6809 versions of X-FORTH, Compatible with FIG FORTH. AND, THERE'S EVEN MORE WE DON'T HAVE ROOM TO LIST!

Supplied on one 8" Disk or 2, 5" disk(s) with a 400 page manual in a hard cover binder. Disk(s) have the source of everything but the core of X-FORTH, which will be available later at extra cost. You get it all!!!

We wanted to provide a better FORTH with more extras at less cost and we did it!!!! Here it is!

All for only

Manual available separately for $49.95, with credit for later purchase. **$149.95**

WHAT ARE YOU WAITING FOR ...

NEW

Exclusively From Frank Hogg . . .

DYNASOFT PASCAL FOR 6809 FLEX™

DYNASOFT PASCAL is now available for **FLEX 9™** systems!

Version 1.3 is an enhancement of 1.2 with:

- upper and lower case source text
- disk based compiler accepts standard text files as input
- ROMable position—independent interpreter less than 1.2K long
- ROMable position—independent p—code
- direct interface to FMS routine
- UNIX* style input/output redirection
- compiler runs in 16K with a single disk drive
- ideal for control applications
- entire run—time source available on disk

Price: Only $59.95. For run-time source add $30.00

™ trademark of TSC * trademark of Bell Labs

See Catalog on Page 9 for Ordering Information . . .

BY JIM SCHREIER

THE BILL PAYER SYSTEM ™

THE BILL PAYER is a package of 10 menu driven programs in TSC Extended Basic. This powerful system helps you keep track of your bills. You can create a vendor list, enter invoices to be paid, generate various reports about them, print your checks and much more. Uses random access files.

THE PURCHASE ORDER system adds purchase orders to the BILL PAYER. This package of programs adds another level of control to your expenditures. Prints out purchase orders and keeps track of purchases.

INCOME/EXPENSE LEDGER. This valuable package is most appreciated at tax time. Allows up to 99 income and expense numbers. Ties into the PURCHASE ORDER system.

Includes manual and source supplied on disk in TSC Extended Basic.

THE BILL PAYER	$ 89.95
PURCHASE ORDER	49.95
INC/EXP. LEDGER	$49.95
	$189.95

BUY ALL THREE AT ONE TIME FOR 169.95 AND SAVE $19.90.

See Catalog on Page 9 for Ordering Information . . .

FRANK HOGG LABORATORY, INC. 130 MIDTOWN PLAZA • SYRACUSE, N.Y. 13210
(315) 474-7856

Dataman

DATABASE MANAGEMENT

MEANS BUSINESS

DATAMAN is made up of 16 menu-driven programs which make up a powerful sequential DBM system under FLEX. You can think of DATAMAN as a computerized card index. Use DATAMAN at home to keep track of your household possesions, record and tape libraries, mailing lists for cards and letters, etc. For your business, use DATAMAN for product inventory, newsletter subscriptions, marketing systems, check balance reconciliation and much more. The uses are limited only by your imagination. DATAMAN turns your 6800/6809 computer into a powerful database management system which can save you a great deal of time and money.

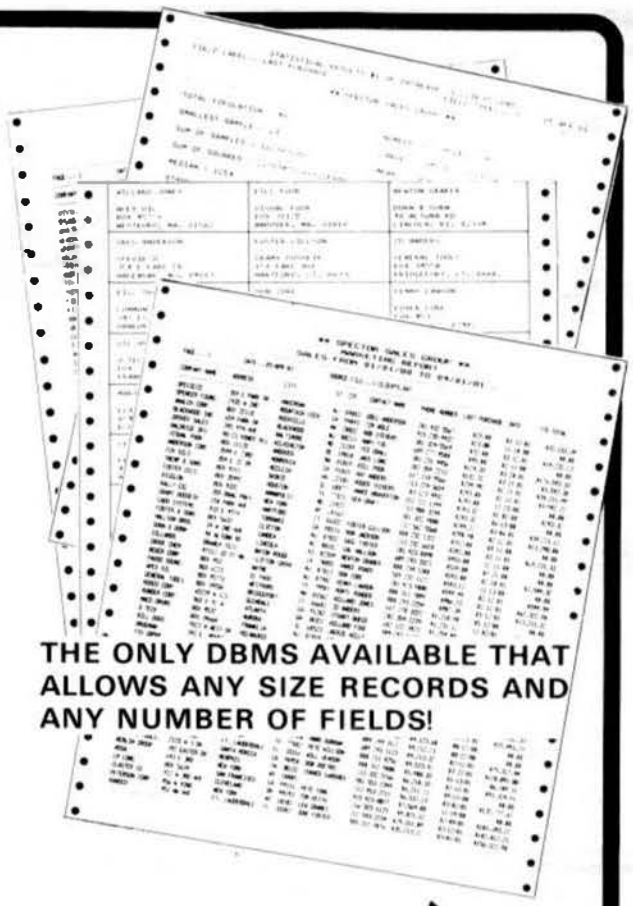
You can create databases and do maintenance. Report writing is easy with both vertical and horizontal formats. Both 80 and 132 column reports are supported in the horizontal format. Formats are saved with run time options available. Label printing is made easy with up to 3 easily changed label drivers on line, and as with other DATAMAN output programs, the output can be spooled to disk for later printing. A statistics package gives up to 24 statistical values. You can transfer records from one database to another, blow away records, even merge two unlike databases together on a key. Sorting on up to 20 keys is done with a sort editor which uses the TSC Sort/Merge package. You can even build 'PR' files for use with the TSC Text Processor for form letters, invoices, reports, etc. The list of features goes on and on!

You can create a database with any number of fields. Most database systems require you to limit the total data in one record to 254 characters or one sector. Not so with DATAMAN! DATAMAN allows your records to be of any length. We made DATAMAN to work for you ..... not against you!

DATAMAN was designed for easy modification and customizing with a "SHELL" program. A "SYSTEM.GEN" file allows almost any terminal or printer to be used with DATAMAN. Comes with every line of source code, program and variable descriptions, and a 160+ page ring bound manual that is second to none!

$149.95

*FLEX is a trademark of Technical Systems Consultants, Inc.



Datarand NEW

RANDOM ACCESS

If you work with databases containing more than 20 records and you need super fast look-up and editing, DATARAND is for you. DATARAND converts DATAMAN files to random files with up to 9 keys. No need to worry about sector blocking or record length. DATARAND takes care of everything! You can look up a record by key value, key wildcard, or record number. DATARAND uses hashed key files which means no slow ISAM searches. Editing is quick and easy using the same screen layouts as DATAMAN.

The same "Record Code" concept is used to make operation simple. In fact, except for the speed, it's hard to tell that you're not in DATAMAN. You can add records to DATARAND just like in the "Append" section of DATAMAN file maintenance. When you need to use DATAMAN, you can copy the desired records to DATAMAN. You can also transfer records from DATARAND which takes care of file reorganization. DATARAND comes with an instruction manual and every line of source code.

$49.95

U.S.A. add $2.50 for Standard UPS Shipping & Handling
Foreign orders add 10% Surface, 20% Airmail
DEALER INQUIRY ENCOURAGED. Contact Frank Hogg for more information.
ATTENTION PROGRAMMERS!
We are looking for quality software to market ... Contact Frank Hogg.

Datacalc Coming Soon!

FRANK HOGG LABORATORY, INC. 130 MIDTOWN PLAZA • SYRACUSE, N.Y. 13210
(315) 474-7856

NEW

By Dick Bartholomew

BASIC PROGRAMMERS TOOLKIT

$49.95 object only
$69.95 with source on DISK!

1... DECOMPIL — Turns .BAC into .BAS files.
2... XREF — Cross reference listing of BASIC programs.
3... EDIT — Edit a BASIC program that's in memory while in BASIC!!!!!

NOTE: For TSC XBASIC only.

NEW

EXTENDED USE UTILITIES PACKAGE

$49.95 object only
$69.95 with source on DISK!

1. CRTSET like TTYSET for CRT's
2. USERINFO adds more than just diskname and number.
3. SCAN list forward and backward thru a file.
4. BROWSE Like SCAN but in memory
5. LOAD offset loader
6. DISKDUMP to any port.
7. INIT memory set command
8. SAVETEXT to disk
9. READTEXT from disk
10. REDIRECT change control port
11. REPLACE prompt memory change
12. MIRROR BACKUP fast sector copy

PASSWORD PROTECTION PACKAGE

$69.95 object only
$89.95 with source on DISK!

Six programs that allow you to protect your system by preventing a bootup without the proper password.

— Warning —

PASSWORD can be overcome if the user has access to the reset button or another disk without password protection that will boot FLEX.

Programs are written in 6809 assembly language.

See Catalog on Page 9 for Ordering Information . . .

From Bud Pass

TABULA RASA

A VISICALC* TYPE PROGRAM

This program provides for the generation and maintenance of tabular computation schemes used for analysis of business, sales and economic scenarios. Simple user interface so even non programmer types can use it.

In TSC Extended Basic. Includes Source on Disk. $100.00

*T.M. PERSONAL SOFTWARE

SUPER SLEUTH

Disassembler System

Examine and/or modify object program files on disk or in memory for 6 00, 6 01, 6 09 ..... $99.00
SUPER SLEUTH Z-80-8080-8085 Version of above $99.00
CROSS-ASSEMBLER for 6800/1, 6805, 6502, Z-80, 8080/5 ..... ea. $50.00

From Dale Puckett

READTEST

This program actually reads a text file that you prepared and tells you how well it was written. READTEST is a must for all writers and writing instructors. Overall index tells who can read it and who would print it. Reports can be submitted with your articles or manuscript. Fast 68XX object code. Runs in Flex.

6800 OR 6 09 ASMB. — $54.95 OBJECT, $74.95 W/SOURCE, ON DISK

ESTHER

The enhanced game of Eliza in fast machine language. Artificial intelligence in pure 68XX code. Great fun. Perfect demo program.

6 00 OR 6809 ASMB. — $39.95 OBJECT, $59.95 W/SOURCE, ON DISK

See Catalog on Page 9 for Ordering Information . . .

From A.R. Boll

READTAPE

READ TRS-80 LEVEL II BASIC TAPES

This program, with an easy to make interface * will read TRS-80 LEVEL II BASIC tapes and convert the programs to TSC BASIC. Those things that can't be converted are flagged so that you can find them easily with the TSC text editor. Now you can use all that TRS-80 software out there.

6809 Assembly language - includes source on DISK! $54.95

*(instructions and schematic included - cost about $2 to build)

PLOT

Now you can have GRAPHICS added to all your programs. Just write the data out to a virtual array and call PLOT. PLOT is written in TSC XBASIC and the source is included on the disk.

INFINITE RESOLUTION GRAPHICS ON YOUR TERMINAL OR PRINTER. HISTOGRAMS, BARGRAPHS, XY PLOTS PLUS OTHERS.

IN TSC XBASIC. SOURCE INCLUDED ON DISK. $44.95

From Peter Murray

JOB CONTROL PROGRAM

This is the most powerful program sold! Why? Because with it you can enhance every other program you own. A procedure file contains input for such calling programs as FLEX, FLEX utility commands and other development software. LIBJCP is used as a Flex command within a procedure to load and execute another procedure. Of all the programs we sell, this is the one we use the most and the one we would least like to do without! What more can I say.

6800 OR 6809

Object Only	$ 49.95
With Source on Disk	89.95
Upgrade from 6800	39.95

See Catalog on Page 9 for Ordering Information . . .

SOFTWARE CATALOG

PROGRAM	LANGUAGE	OBJECT	W/SOURCE ON DISK
X-FORTH	6809/6800		***$149.95
Datamon	TSC XBASIC		$149.95
Datorond	TSC XBASIC		69.95
*Bill Payer	TSC XBASIC		89.95
*Purchase Order	TSC XBASIC		49.95
*Income/Expense	TSC XBASIC		49.95
All Three	TSC XBASIC		169.95
Basic Prog. Toolkit	6809 ASMB	$69.95	69.95
Password Protection	6809 ASMB	69.95	89.95
Extended Utilities	6809 ASMB	49.95	69.95
Job Control Prog.	6800/6809 ASMB	49.95	89.95
Esther	6800/6809 SMB	39.95	39.95
Readtest	6800/6809 ASMB	54.95	74.95
Help	6800/6809 ASMB	29.95	49.95
Dynasoft Pascal	6809	59.95	** 89.95
Plot	TSC XBASIC		44.95
Read TRS80 Tapes	6809 ASMB		54.95
Super Sleuth	6800/6809		99.00
Z80 Super Sleuth	6800/6809		99.00
Cross Assemblers	MACROS FOR TSC 6809 ASMB		EA. 49.95
	6800/1, 6805, 6502, Z-80, 8080/5	3 for	99.95
Mailing List	TSC XBASIC/6809		99.95
Format Display	TSC XBASIC/6809		49.95
Tabula Rasa	TSC XBASIC		100.00

** SOURCE AND REPRODUCTION LICENSE-RUNTIME ONLY.
***Includes everything but the core.

U.S.A. add $2.50 for Standard UPS Shipping & Handling
Foreign orders add 10% Surface, 20% Airmail.

Specify 5" or 8" size disk and if for 6800 or 6809 system.

OUR SOFTWARE IS CMX *COMPATIBLE.*

OS-9 VERSIONS TO COME.

FRANK HOGG LABORATORY, INC.

130 MIDTOWN PLAZA • SYRACUSE, N.Y. 13210
(315) 474-7856

DISK DRIVES - $AVE - $AVE

☆ Limited Quantity ☆

A Special Purchase for 68' Micro Journal Readers Only!

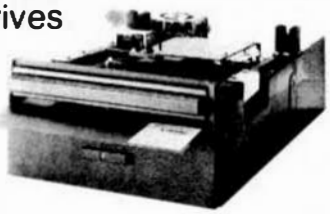
SAVE HUNDREDS OF $

FACTORY FRESH — LATEST PRODUCTION — BRAND NEW

LIMITED OFFER

Remex RFD4000

8" Floppy Disc Drives
Double sided . . .
Double density! !



$498.50

2 for $975.00
Add: $7.50 each - Shipping and Handling
Offers quality and features found in drives costing much more! ■ Single or Double Density ■ Double-sided Drive ■ Door Lock INCLUDED ■ Write-Protect INCLUDED ■ 90 Day Warranty ■ Compatible with Shugart 850 ■ Low Power Operation ensures LONGER LIFE!!

68 Micro Readers Only!

Single Disk System Copy Routine in Assembled Source Included for a Complete 'Single Drive' Disk System

DUAL 8" DISK DRIVE CABINET WITH POWER SUPPLY AND CABLE $275.00

DUAL 5" DISK DRIVE CABINET WITH POWER SUPPLY AND CABLE $225.00

USA Add $10.50 for Shipping and Handling.
Add 10% Foreign Surface Add 20% Foreign Airmail

CT-64 Bare Board Set $59.95
Includes: Complete Documentation, Memory Board, Serial Board, Screen Read Board, Main Terminal Video Board. These are the last ones available anywhere. Limited Quantity.

Dealer Inquiries Invited

Cannot guarantee supply will last if paid for by personal check as personal checks require 3-4 weeks to clear - Recommend Money Order or Certified Bank Check.

SOUTH EAST MEDIA

P.O. Box 794 Chattanooga TN 37443
1-615-877-2241

WE SELL ONLY DATALIFE DISKS

Flex User Notes

BY: RONALD W. ANDERSON
3540 STRUBRIDGE COURT
ANN ARBOR, MI 48105

HELP!

I'm suddenly feeling quite overwhelmed by responses to my column, not from hobbyist readers, but from supplier readers. Ray Talbot saw my saga of implementing FORTH from the Fig documentation, and sent me a copy of tFORTH+ and his documentation. After some difficulties (mine) in reading the disks he sent me, I got tFORTH up and running. I have to report that Ray has solved many of the difficulties about which I wrote. He has come up with an interesting approach of a "hybrid disk" (my nomenclature). He creates a system disk on which the first 160 sectors are Flex compatible. These contain FORTH.CMD, FLEX.SYS, and any other utilities the user cares to have on the disk. The remaining sectors are standard FORTH screens. On a standard 35 track single density, single sided disk, there are 350 sectors. A screen uses 4 sectors, so that there is room for 87 screens on the disk. With Ray's split of the disk, the first screen is #40. The system disk can therefore hold 47 screens. tFORTH+ uses all of them for such things as Disk utilities, tools (for documentation of FORTH screens), ARRAYS, CASES, the ASSEMBLER, a screen EDITOR and the standard FORTH line EDITOR, DOUBLE PRECISION MATH, and a few other features.

* Ray Talbot supplies the tFORTH advertized and sold by Kenyon Microsystems in Houston, TX. Kenyon has undertaken the job of marketing for Ray, who is presently located in Riverside CA. Ray sent me a letter and some copies of correspondence regarding benchmark times on a Prime Number program (again precipitated by the Moreira article in '68'). With some improvements suggested by the copies, I was able to get my Prime program to run in tFORTH right up there with the fastest of them. More on this some time when we do a comparison again of some compilers. At any rate, the documentation with tFORTH mentions the REDEFINED message very early, and indicates that it is a warning and not an error. (Would have saved me a week of hair pulling a few months ago). The documentation is useful, and several grades above what I had previously, but still lacking in explainations of the "advanced features" such as the ARRAYS and CASES added to FORTH in the tFORTH+ version.

I know Ray is going to write me that the definitions are right there on the disk with comments. Somehow, I need more than four word comments to explain the use of those CASES to me, although I was able to grasp the ARRAY features quickly and in fact use them to clean up my Prime program significantly. I must be gaining, because the modified Prime program ran correctly the first try, and I had made several non-trivial modifications in it. I am still not ready for an 'I LIKE FORTH' bumper sticker, but I'm beginning to feel a bit more comfortable with it.

PASCAL COMPILER FROM TSC

I've seen TSC's new Pascal compiler. It is the fastest (except for floating point calculations), but it uses the largest runtime package, and it has very significant deviations from the Jensen and Wirth standard. A standard Pascal program requires from minor (comment out the first line) to major (add the function CONV( ) to all integer variables in mixed mode arithmetic expressions) doctoring for it to be accepted by the TSC compiler. Of course the compiler works flawlessly when the program has been "adjusted" to suit it. We have All come to respect TSC for their capabilities, and this is a capable compiler. I think however, that it is stretching the point to call it a Pascal compiler. Come on TSC, let us leave the first line intact, do the integer to REAL conversions automatically in mixed mode arithmetic, and fix the funny with strings. Then you will have a Genuine Pascal compiler. The others will compile a program with no changes whatever.

LATEST PASCAL FROM LUCIDATA

Lucidata has completed their release 3.9 with all the additions advertized (Scientific functions, dynamic variables, and some nice extensions). They have also had a change in policy. In order not to get it wrong, I quote from their customer agreement form. "I understand that any application software written by me using Lucidata products and requiring them to function, may be supplied to third parties provided that the Lucidata product is "bound" with my software and supplied only as a single binary command file. Any other situation requires the execution of a License agreement with Lucidata Ltd., or purchase of the product by the third party." Lucidata has a version of their RUN command that loads the user program and the runtime package and then tells you the starting, ending and transfer addresses so that you may SAVE the program as a single runnable binary file. It is this process that they call "binding" the user program and thier runtime package, and such a bound program is now not restricted with regard to transfer to a third party.

Incidentally, to my knowledge, Lucidata is the only supplier of Pascal presently supporting the 6800 processor. This new version (3.9 for FLEX9 and 3.2 for FLEX2) is available for both processors. The '09 version has been optimized to use the '09 instruction set, and it runs about twice as fast as the '00 version.

MURPHY AT WORK

I should know better than to try writing this during a thunderstorm. Power went down for a half second brownout, and my terminal lost touch with the computer. Had to work a bit to save most of the file from memory where some of it got scrambled. Maybe someday I'll have a standby power unit for such times! Having had the power down reminds me that I've been wanting to let my readers in on two additions to Murphy's laws that I've come up with. The first is Anderson's law of lost tools. It is stated as follows:

You will not find the tool you are looking for today. You will instead find the tool you were looking for last week, but only if you have meanwhile bought a replacement for it.

The second is called "The law of simultaneous emergencies", the popular statement of which is 'Why does everything go wrong at once?'. Actually it goes deeper than that. It is stated:

The probability of an emergency occurring today is directly proportional to the number that have already occurred.

There is a corolary to this last law, called the unevenness rule. It is stated:

If you have 9 projects all stalled for the lack of one major part in each, all the missing parts will arrive on the same day. There is little use trying to calculate the probability of this happening since it is almost a sure thing.

While I'm at it, I'd like to add just one more. It is called "Anderson's good idea law". It is:

The best idea for the solution to the problem at hand

will occur the day after the project is shipped. — Corolary: When this happens, the probability of repeating the project will automatically go to zero so that the good solution cannot be used. Enough of that for now? Anyone else out there with some good ones?

TRIGONOMETRIC FUNCTIONS

I've lately seen some BASIC's and a Pascal with "extanded Trig functions", which generally include the three not normally included in BASIC. These are Tangent, Arcsin and Arccos. If you ever need these, (I have on two occasions), they are easily generated from the others. The Pascal versions would be as follows:

```
FUNCTION TAN (X:REAL):REAL;

BEGIN
  TAN := SIN(X)/COS(X);
END;


FUNCTION ARCCOS (X:REAL):REAL;

BEGIN
  ARCCOS := ARCTAN (SQRT((1-SQR(X))/SQR(X)))
END;


FUNCTION ARCSIN (X:REAL):REAL;

BEGIN
  ARCSIN := ARCTAN (SQRT(SQR(X)/(1-SQR(X))))
END;
```

The latter two are based on the fact that SQR (SIN(X)) + SQR (COS(X)) = 1. That is, for any angle, the square of the Sine plus the square of the Cosine is equal to 1. Further, the Sine/Cosine is the TANGENT, which we need if we are to use the Arctan function. The same sort of thing can be done defining functions in BASIC, of course, but the naming of the functions is not quite as handy. Note that in Pascal, SQR is the square function, and SQRT is the square root function.

NCC SHOW IN CHICAGO

Assuming that I am farther ahead than most everyone else who is writing for '68', I guess you have all read about NCC by now, but here's my report from a point of view probably different from that of the others. On Monday night prior to NCC I received a call from Richard Don of Gimix, who asked if I planned to attend. He invited me to his home for dinner on Sunday evening, indicating that there would be some people there whose software I had evaluated. I asked "do I dare?", and we decided that I was probably big enough to defend myself adequately. You may not realize that a favorable review from me may have been preceeded by several conversations with the supplier regarding what I see as bugs. As I mentioned in a previous column, the suppliers come through and make improvements in their products. However, I am not always very diplomatic in my critiques, and I certainly have managed to offend some of the authors.

With this background, you might imagine that I had some reservations about being surrounded by the folks on whom I have been very hard. I accepted the invitation and arrived at the Northbrook Sheraton at about 5:30 P.M. on Sunday. Shortly, I received a call from Don Williams asking if I wanted to ride in his van over to Richard Don's for dinner. I accepted and we headed off in a group for Richard's house.

The evening was most pleasant. Richard is a great host. How he managed to keep everyone's name straight, I don't know. There were probably at least a dozen people there that he had never met previously. I had extended conversations with Bob Bundy (author of Stylograph), Ray Talbot (6809 implementation of Fig FORTH, and Kenyon Microsystems tFORTH), Al Jost (Dynasoft Pascal), Tom Crosley (PIE), and a few others. I met and spoke to Ken Kaplan (OS9), Dick Bartholemew (implementor of UCSD Pascal for 6809), and Dan Farnsworth, who sells 6800/09 based business systems in Florida, writing software for many business applications in Assembler.

I was impressed by the GIMIX booth, at which several systems were running OS9, FORTH, PIE, Stylograph, and a slick color graphics demonstration. It appears that much good Flex and OS9 compatible software is available in a version that will run on GIMIX hardware. The show itself was overwhelming in size, mostly the larger manufacturers and their large computers and peripherals. GIMIX and Smoke Signal were about the only SS-50 suppliers represented. It was a thoroughly enjoyable experience.

READER RESPONSES

My May '68' arrived in the mail while I was away for the show. With it were three letters, two of which I would like to comment on. They both dealt with my "Challenge" in the May column. It seems that my lack of an advanced degree, or perhaps the antiquity of my Engineering education (I call it B.C. for Before Computers), has left me unaware of some areas of mathematics. Peter Stark sent me copies of several pages of his book "Introduction to Numerical Methods" published by Macmillan Company. It would seem that there is a rather straightforward method for calculating the best coefficients for a truncated series approximation. Peter has written about it in his book. It involves the use of Chebyshev Polynomials, to which I had not been exposed in my Engineering math courses.

The second letter made some comments regarding my criticism of FORTH, probably well deserved, and then indicated that there are available solutions to the error minimization problem that I had posed. The writer missed my point just a bit. He said "The U.S. Department of Commerce 'Handbook of Mathematical Functions' gives a number of the approximations which you want. These might be easier than trying to duplicate the work." I thought I had made it clear that I had the approximations that I needed, but thought you might find it enjoyable to look at these approximations and perhaps find some ways to arrive at them.

This might have been a rather dumb project since the techniques for solving it analytically are known and trial and error solutions are not required, (though I didn't know that when I posed the problem). I've since then received a couple of other letters from readers with sets of coefficients for these functions. Most all have said exactly the same thing, that these solutions are avilable in standard reference books.

Though I must plead ignorance in this case, I must say that in general I don't think re-inventing the wheel is necessarily a waste of time. I frequently start from Newton's laws and derive a solution to a problem that I am working on, knowing that it has been solved before. Frequently the equations that I derive (or more usually the act of deriving them) gives me further insight into the problem, that is to say, I get more than a solution out of solving it for myself. One of my recent wheel re-invention escapades resulted in an insight that reduced the electronics that had previously been used by about 50%, and at the same time, revealed an approach that may be used to solve other related problems. I think the act of learning always involves some re-invention. After all, in school, we spend much of our time solving problems that have already been solved.

I received Creative Computing for May today, and find support for my position. See Pg. 66. Some people build scale models of the Brooklyn Bridge using 25,000 toothpicks. If we computer hobbyists find pleasure and relaxation in re-inventing Sort algorithms, Random Number generators, Trig Function approximations, or Prime number finding algorithms, why not do it? At any rate, I've ended up learning a technique that I didn't know existed before. Thanks to all "answerers" for the information.
Oh yes, I almost forgot. Seems that a few people missed my disclaimer about my own spelling, and jumped on me for a couple of dumbs in my paragraph regarding spelling in the May issue. It seems that I spelled the abbreviation etc. properly, but misspelled et cetera the latin words for which it is an abbreviation. An anonymous post card informed me of this, adding that "—this is elementary high-school-level Latin". Sorry fella, I took Spanish. Oh, by the way "Kudzu" if you are going to be critical, how about signing your real name? Besides, you missed my other favorite that I have been spelling wrong (but consistently). I've been using "persue" rather than the correct pursue. Sorry about that.

Perhaps I should take this opportunity to jump on Don Williams a bit, since he didn't publish the listing for my "Puzzle" or for the Sine series problem. In addition, somehow his text processor changed my up arrow so that X(up arrow)3 became XCTRL-3 which makes no sense at all. In other places in the same text it became Xpwr3 which does make sense.

Ed's Note: * Recently tFORTH and associated software has been returned to Dr. Ray Talbot for his personal attention and marketing. (See advertisement back cover - this issue) I personally feel that this is a good move, as Dr. Talbot is a 'real' pioneer in this particular field and is one of the world's foremost authorities' on FORTH. I have heard nothing but GOOD reports from those using FORTH.

As to the spelling errors: It is and will continue to be the policy of 68 Micro Journal to keep things as "ACCURATE AND INFORMAL" as possible. This implies that occasionally you will not only find words that are spelled wrong but you will also find grammatical errors sprinkled about. You will have to admit, 68 Micro Journal has experienced LESS errors in technical content than most any other computer magazine, bar none. My instructions are: If it makes sense, leave it alone! My staff could correct many small and non-important spelling and code errors, but in some cases they could change the intent or meaning of material that we received on disk text files, printer print outs and other sources. Soooo..we leave most of the nik-pik stuff alone, unless it would devaluate the useful application of the subject material.

UNDERSTANDING SUBROUTINES PART 3 — ADVANCED TOPICS

John F. Wakerly
Micro Systems Engineering
257 Castro Street, Suite 2E3
Mountain View, CA 94041

John Wakerly is an independent consultant and a consulting associate professor at Stanford University. This three-part tutorial on subroutines in Pascal and 6809 assembly language is adapted from his recently-published book, Microcomputer Architecture and Programming, *copyright 1981, with permission of the publishers, John Wiley & Sons, Inc. (The book is also available directly from the author at MSE Books; see advertisement elsewhere in this issue.)*

In the first two parts of this series we discussed subroutines and parameters in Pascal and in Motorola 6809 assembly language. Now we continue with the advanced topics of recursion and coroutines.

RECURSIVE PROCEDURES AND FUNCTIONS

A procedure or function that calls itself is said to be *recursive*. The Pascal factorial function from Table 8 in Part 1 of this series is redefined below as a recursive function:

```
FUNCTION Fact (i : integer) : real;
  BEGIN
    IF i <= 1 THEN Fact := 1
    ELSE Fact := i * Fact(i-1);
  END;
```

Essential to this definition is a *basis part* that defines Fact(i) to be 1 for any i<=1. For larger i, Fact(i) is defined to be the product of i and Fact(i-1). For example, to compute Fact(5) we must first compute Fact(4), which depends on Fact(3), which depends on Fact(2), which depends on Fact(1). The basis part ensures that we eventually reach a value of i for which Fact(i) does not depend on Fact(i-1), so that we can eventually terminate the recursive calls of Fact.

The example above illustrates *simple recursion*, using a procedure that calls itself directly. It is also possible for a procedure to call one or more intermediate procedures that eventually call it. This is called *indirect recursion* and is illustrated below.

```
PROCEDURE ProcA(x, y : integer);
  BEGIN

    ...
    ProcB(a); {Call ProcB}
    ...
  END;

PROCEDURE ProcB(z : integer);
  BEGIN
    ...
    ProcA(b,c); {Call ProcA}
    ...
  END;
```

Since Pascal requires a procedure to be defined before it is called, the above program fragment is syntactically incorrect as it stands. The programmer must inform the compiler of the forward reference by placing the following declaration before the definition of ProcA:

```
PROCEDURE ProcB(z : integer); forward;
```

The directive "forward" takes the place of the block that is normally required in the syntax of Figure 3 in Part 1. It alerts the compiler that the block defining ProcB is coming later. The parameter list is included in the forward declaration so that statements that refer to ProcB can be checked and compiled. Later, body of ProcB may be defined in the normal way, except that the parameter list is not repeated.

Block-structured languages such as Algol and Pascal allow all procedures and functions to be called recursively. Unstructured languages like Fortran usually do not permit recursion. Some BASICs allow recursion, others do not.

The recursive function definition above may be clever, but the iterative solution in Table 8 in Part 1 may be more efficient. In general, problems that have easily-stated iterative solutions are best solved iteratively. Recursion should be re-

served for problems that are most clearly stated recursively or that huve no obvious iterative solution. An example of such a problem is given in the next section.

RECURSIVE SUBROUTINES

Recursion can be utilized in assembly language subroutines, but it places constraints on the subroutine calling and parameter passing conventions that may be used. Return addresses, parameters, and local variables may *not* be stored in dedicated, static locations, because they would be wiped out the first time that the subroutine recursively called itself. Instead, a new area for the return address, parameters, and local variables must be allocated on each recursive call, and deallocated on each return. Hence, a pushdown stack is the appropriate data structurc for storing these items.

A subroutine that stores its return address and all parameters and local variables using a stack convention such as the one in Part 2 can be called recursively without error. This explains why Pascal procedures can call each other recursively, and Fortran subprograms cannot: Fortran normally uses static memory allocation for parameters.

A pair of recursive subroutines can be used to analyze the game of NIM, a two-person game that begins with a heap of sticks. The players alternately remove sticks from the heap; the player who removes the last stick loses. The game is fully characterized by two parameters: NHEAP is the number of sticks initially in the heap, and NTAKE is the maximum number of sticks a player may take on each turn, the minimum being 1.

We would like to write a program that determines, given NHEAP and NTAKE, whether or not an intelligent first player (P1) can always win by making optimal moves. In order to formulate a recursive algorithm to make this determination, we first define a *winning position* for P1:

(1) If it is P1's turn and there are no sticks left, then the second player (P2) has just taken the last stick. This is a winning position for P1.

(2) If it is P1's turn and there is at least one winner among the new positions obtained by taking 1 to minimum(NTAKE,STICKSLEFT) sticks, then P1 can take the appropriate number of sticks and eventually win. This is a winning position for P1.

(3) If it is P2's turn and there are no sticks left, then P1 has just taken the last stick. This is *not* a winning position for P1.

(4) If it is P2's turn and at least one of the new positions obtained by taking 1 to minimum(NTAKE,STICKSLEFT) sticks is not a winner, P2 can take the appropriate number of sticks to keep P1 from winning. This is *not* a winning position for P1.

Steps 1 and 3 above form the basis parts of two recursive subroutines, P1TURN and P2TURN, that call each other. Each subroutine determines, given NTAKE and STICKSLEFT, whether or not the current position is a winning position for P1, assuming it is P1's or P2's turn to move. The subroutines are coded in 6809 assembly language in Table 1. Input and output parameters are passed in registers, and local variables are saved in the stack at the beginning of each subroutine and restored on exit. A program can initialize NTAKE to any desired value and call P1TURN with the initial heap size in register A to determine whether or not the game is a guaranteed win for an intelligent first player, as in the example below.

```
        LDA  #5         Take 5 sticks maximum at a time.
        STA  NTAKE
        LDA  #30        Can I win starting with 30 sticks?
        JSR  P1TURN
        BEQ  IWIN
ILOSE   ...
        ...
IWIN    ...
        ...
```

Recursive programs often perform a tremendous amount of useful computation with relatively little memory. For example, the NIM subroutines are short, they have only one global variable (NTAKE), and they never have more than about 4·NHEAP bytes on the stack. Yet called with NHEAP=30 and NTAKE=5, the two subroutines are executed a total of 1,687,501 times. Try to figure out whether P1 won or lost that game yourself!

COROUTINES

So far we have discussed subroutines in the context of a master/slave relationship — a calling program (master) calls the subroutine (slave), which executes from beginning to end and returns to the calling program. In Pascal, subroutines (procedures and functions) are so subservient that they aren't even allowed to remember their own local data between successive calls. *Coroutines* replace this master/slave structure with a set of cooperating program modules with no identifiable master. Consider the following problem statement by R. W. Floyd*:

> Read lines of text, until a completely blank line is found. Eliminate redundant blanks between the words. Print the text, thirty characters to a line, without breaking words between lines.

This problem statement describes the operation of a simple text formatter. According to Floyd, novice programmers take an unreasonably long time to solve this problem using typical programming languages. Even though both input and output are naturally expressed using levels of iteration, the input and output iterations do not mesh, which can make controlling the input and output an "undisciplined mess."

The problem can be solved naturally by decomposing it into three communicating coroutines for reading input characters, assembling them into words, and printing words, as

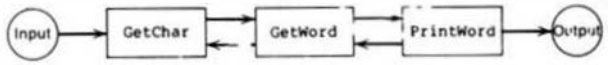


FIGURE 1 Three coroutines for text formatting.

Figure 1. The GetChar coroutine reads input characters and detects blank lines. GetWord assembles words and discards spaces, getting individual characters from GetChar and passing complete words to PrintWord. The PrintWord coroutine formats words onto lines according to the the line length limit.

EXTENDED-PASCAL COROUTINES

In order to study coroutines in more detail, we shall extend the syntax of Pascal to include coroutines. We'll use a new reserved word "COROUTINE" to define coroutines and a reserved word "RESUME" to call a coroutine. When a coroutine is "resumed" for the first time, execution is started at its first statement. Once entered, a coroutine Cor1 may be temporarily suspended by the statement "RESUME Cor2", which transfers control to Cor2, another coroutine. Now the statement "RE-

SUME Cor1" will leave Cor2 and continue execution of Cor1 at the point just after Cor1 called Cor2, not back at the beginning. Table 2 illustrates.

Table 3 defines the coroutines GetChar, GetWord, and PrintWord for formatting text.** An important difference between coroutines and standard Pascal procedures is that coroutines must preserve the values of their local variables between successive calls. Thus blankLine in GetChar "remembers" whether the line so far has been blank, and column in PrintWord remembers the current output column number in order to properly handle the next word.

Each of the coroutines in Table 3 has been written independently as if the other coroutines were its subroutines. For example, GetChar reads characters and passes them to GetWord; it also translates an end-of-line condition into a space character for GetWord. Looking from another point of view, GetWord calls GetChar from two different places to get a character, totally unaware that GetChar may actually be resumed in either of two different places.

Coroutines GetChar and GetWord contain endless loops, and may appear to never terminate. However, GetChar passes a blankLine flag up to PrintWord, which eventually returns control to the main program.

ASSEMBLY LANGUAGE COROUTINES

In order to program coroutines in assembly language, we need to save a "resumption address" for each coroutine. When Cor1 resumes Cor2, it should save the current value of the program counter in a memory location RES1 and jump to the address contained in a memory location RES2. Now Cor1 may be resumed by jumping to the address that was saved in RES1.

If a coroutine Cor1 in the 6809 calls Cor2 by JSR COR2 and vice versa, then the following statements may be used to link the two coroutines:

```
COR1   PULS Y        Save Cor2's resumption address
       STY  RES2       in RES2.
       JMP  [RES1]   Jump to Cor1's resumption address.
COR2   PULS Y        Save Cor1's resumption address
       STY  RES1       in RES1.
       JMP  [RES2]   Jump to Cor2's resumption address.
RES1   RMB  2        Storage for Cor1's resumption address.
RES2   RMB  2        Storage for Cor2's resumption address.
```

Notice that the JMP instructions use indirect addressing. All that remains is for the values stored in RES1 and RES2 to be initialized when the program is started, to the address of the first executable instruction of each coroutine.

The line-formatting coroutines in Table 3 have been coded for the 6809 in Table 4. A macro COLINK is defined at the end of the program to generate coroutine linkages. In general, the coroutine linkage instructions must take into account both the coroutine that is being suspended and the one that is being resumed. For example, GetWord can be resumed from both GetChar and PrintWord and so two different linkages are needed. However, notice that there is still only one resumption address for each coroutine.

COROUTINE APPLICATIONS

Coroutines find their most common application in programs that read inputs, perform some transformation, and produce outputs, as shown in Figure 2(a). Because of the analogy with electronics, such programs are often called *filters*; sometimes filters are cascaded. For example, the following filters might be applied to a text file to find spelling errors:

(1) Remove all punctuation and reformat the text so that each line contains only one word.

(2) Remove all words that consist of only upper case letters (assuming that they are acronyms or mnemonics).

(3) Translate each upper case letter into the corresponding lower case letter.

(4) Look up each word in a dictionary and output all words that are not found.

A program could be devised to perform these tasks one at a time, producing three temporary files that pass the results of one filter to the next, as shown in Figure 2(b). Alternatively, the program could be organized as four coroutines as shown in Figure 2(c). In the first case, the individual filters can be executed at different times and therefore can be fit individually into a small memory. In the second case, the coroutine structure avoids the extra file space and processing time associated with reading and writing temporary files, at the possible expense of requiring a larger program memory.

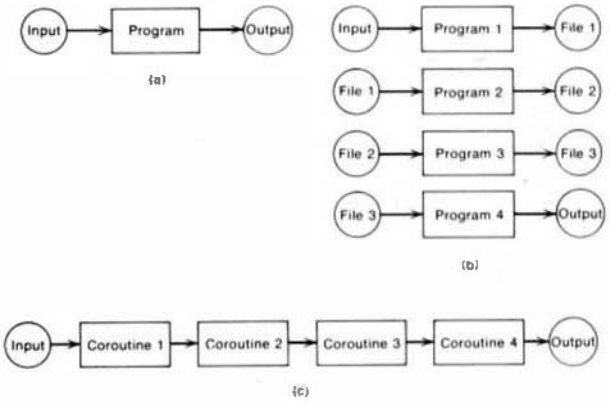


FIGURE 2 Filters and coroutines: (a) a simple filter; (b) a cascade of filters using intermediate files; (c) an equivalent coroutine structure.

TABLE 1 Recursive subroutines to analyze the game of NIM.

```
* Subroutine P1TURN determines if the current position
* is a winner, given NTAKE (a global variable) and
* STICKSLEFT (passed in register A), assuming that it is
* P1's turn to move.  P1TURN saves registers A and B on
* entry, restoring them on exit.  The result is returned
* in condition bit 2: 1 if a winning position, else 0.
*
P1TURN PSHS A,B      Save registers B and A on stack.
       TST  A        Any sticks left?
       BEQ  WIN      Return with Z=1 if none (we won!).
       LDB  NTAKE    B := maximum # of sticks to take.
       BRA  P1L2     Jump into loop.
P1LOOP JSR  P2TURN   Do we have a winning position?
       BEQ  WIN      Found one, mark this a winner.
       DECB          Otherwise, try to take a stick.
       BEQ  LOSE     Lose if we've tried NTAKE sticks.
P1L2   DECA          Also lose if no more sticks left.
       BGE  P1LOOP
*
LOSE   PULS A,B      Restore A and B from stack.
       CLRZ          Return with Z=0 (not a winner).
       RTS
WIN    PULS A,B      Restore A and B from stack.
       SETZ          Return with Z=1 (a winner).
       RTS
```

```
* Subroutine P2TURN determines if the current position
* is a winner, given NTAKE (a global variable) and
* STICKSLEFT (passed in register A), assuming that it is
* P2's turn to move.  P2TURN saves registers A and B on
* entry, restoring them on exit.  The result is returned
* in condition bit Z: 1 if a winning position, else 0.
* Exit code is shared with P1TURN.
*
P2TURN PSHS A,B        Save registers B and A on stack.
       TST  A          Any sticks left?
       BEQ  LOSE       Return with Z=0 if none (we lost).
       LDB  NTAKE      B := maximum # of sticks to take.
       BRA  P2L2       Jump into loop.
P2LOOP JSR  P1TURN     Do we have a losing position?
       BNE  LOSE       Found one, mark this a loser.
       DECB            Otherwise, try to take a stick.
       BEQ  WIN        Win if we've tried NTAKE sticks.
P2L2   DECA            Also win if no more sticks left.
       BGE  P2LOOP
       BR   WIN
*
NTAKE  RMB  1          Max # of sticks to take (global).
```

"The Paradigms of Programming," *Comm. ACM*, Vol. 22, No. 8, August 1979, pp. 455-460.

TABLE 2 Two coroutines.

```
CORDUTINE Cor1;                    CORDUTINE Cor2;
  BEGIN                              BEGIN
    ...                                ...
    RESUME Cor2;                       ...
    ...                                RESUME Cor1;
    ...                                ...
    RESUME Cor2;                       ...
    ...                                ...
    ...                                RESUME Cor1;
    RESUME Cor2;                       ...
    ...                                RESUME Cor1;
  END;                               END;
```

TABLE 3 Line-formatting program using coroutines.

```
PROGRAM Format (input,output);
{ This program reads lines of input text until a
  completely blank line is found. It eliminates extra
  spaces between words and then packs them on output
  lines with a maximum line length of 30 characters,
  never breaking a word in the middle. Words longer
  than 30 characters are truncated.
}
CONST lineLen = 30;
VAR inChar: char; {Pass chars from GetChar to GetWord}
  wordBuf: ARRAY [1..lineLen] OF char; {Accumulates
       words and passes them from GetWord to PrintWord}
  wordPnt: integer; {Index of last valid char in wordBuf}
  blankLine: boolean; {Set true when blank line is read}

COROUTINE GetChar;
  BEGIN
    REPEAT {forever}
      blankLine := true; read(inChar); {'read' sets ...}
      WHILE NOT eoln DO  {... eoln true at end of line.}
        BEGIN
          blankjline := false;
          RESUME GetWord;
          read(inChar);
        END;
      {A space is needed to flush last word on a line.}
      inChar := ' '; RESUME GetWord;
    UNTIL false;
  END;

COROUTINE GetWord;
  BEGIN
    REPEAT {forever}
      wordPnt := 0;
      REPEAT {Skip spaces.}
        RESUME GetChar;
      IF blankLine THEN RESUME PrintWord;
    UNTIL inChar <> ' ';
    REPEAT
      IF wordPnt < lineLength THEN
        BEGIN
          wordPnt := wordPnt + 1;
          wordBuf[wordPnt] := inChar;
        END;
      RESUME GetChar;
    UNTIL inChar = ' ';
    RESUME PrintWord; {Got a word, go print it.}
  UNTIL false;
END;

COROUTINE PrintWord;
  VAR column, i : integer;
  BEGIN
    column := 0;
    RESUME GetWord; {Get first word.}
    WHILE NOT blankLine DO
      BEGIN {Read and print a word.}
        {Will the word fit, including an extra space?}
        IF column = 0 THEN {Do nothing.}
        ELSE IF column+wordPnt+1 <= lineLength THEN
          BEGIN write(' '); column := column+1 END
        {Start a new line if word doesn't fit.}
        ELSE BEGIN writeln; column := 0 END;
        FOR i:=1 TO wordPnt DO {Print the current word.}
          BEGIN write(wordBuf[i]);
                column := column+1 END;
        RESUME GetWord; {Get next word.}
      END;
    writeln; {Finish last line and return to Main.}
  END;
BEGIN {Main Program}
  Printword;
END.
```

TABLE 4 6809 version of line-formatting program.

```
        ORG  $2000
SPC     EQU  $20        ASCII space.
CR      EQU  S0D        ASCII carriage return.
LINELN  EQU  30         Maximum output line length.
WRDBUF  RMB  LINELN     Word buffer.
BLANK   RMB  1          Blank-line flag.
COLUMN  RMB  1          Output column number.
XTEMP   RMB  2          Temporary storage for X.
STACK   RMB  20         Stack area.
STACKE  EQU  *          Stack initialization address.
*
*       COROUTINE GetChar -- returns a character in A.
GCHRIN  LDA  #SFF       Assume we have a blank line unless
        STA  BLANK        we get a nonspace.
GCHR1   JSR  READ       Read a character.
        CMPA #CR        Is it the end of line?
        BEQ  GCHR5
        CLR  BLANK      No, not a blank line.
GCHR2   JSR  GETWRDG    Give the character to GETWRD...
        BRA  GCHR1      ...and do some more.
GCHR5   LDA  #SPC       At end of line, force a space...
        JSR  GETWRDG    ...and give it to GETWRD.
        BRA  GCHRIN     Go read more lines.
*
*       COROUTINE GetWord -- puts a word in WRDBUF[1..X].
GWRDIN  LDX  #0         Set index before start of WRDBUF.
GWRD1   JSR  GETCHR     Get a character.
        TST  BLANK      Hit a blank line?
        BEQ  GWRD2      No, continue.
        JSR  PRTWRD     Yes, resume PRTWRD.
GWRD2   CMPA #SPC       Skip over spaces.
        BEQ  GWRD1
GWRD3   CMPX #LINELN    Is there room left in WRDBUF?
        BHS  GWRD4      No, ignore character.
        LEAX 1,X        Yes, bump X to next buffer index...
        STA  WRDBUF-1,X  ...and put the char into WRDBUF.
GWRD4   JSR  GETCHR     Get another character...
        CMPA #SPC       ...and continue processing until a
        BNE  GWRD3        space character is found.
        JSR  PRTWRD     Now we have a word, go print it...
        BRA  GWRDIN     ...and then get some more words.
```

```
*       COROUTINE PrintWord--prints word in WRDBUF[1..X].
PWRDIN CLR  COLUMN   Set output column to zero.
PWRD1  JSR  GETWRDP  Get a word.
       TST  BLANK    Hit a blank line?
       BNE  PWRD9    Yes, exit.
       TST  COLUMN   No, are we in the middle of a line?
       BEQ  PRTBUF   Print word now if we're at column 0.
       TFR  X,D      Else get the word length (A,B := X)...
       ADDB COLUMN   ...plus the number of characters so far.
       CMPB #LINELN-1 Will word fit, including a space?
       BHI  PWRD5    Start a new line if it won't fit.
       LDA  #SPC     Otherwise output a space...
       JSR  WRITE
       INC  COLUMN
       BRA  PRTBUF   ...and print the word.
PWRD5  JSR  WRITELN  Print CR and LF for a new line.
       CLR  COLUMN
PRTBUF STX  XTEMP    Print the word in WRDBUF.
       LDY  #1
PRTB1  CMPY XTEMP
       BHI  PWRD1    Go process more words when done.
       LDA  WRDBUF-1,Y Else print another character...
       JSR  WRITE
       LEAY 1,Y      ...bump Y to next buffer index...
       INC  COLUMN   ...and update column number.
       BRA  PRTB1
PWRD9  JSR  WRITELN  Print CR and LF for a new line.
       RTS           Return to main program.
*
MAIN   LDS  #STACKE  Initialize SP.
       LDX  #GWRDIN  Initialize coroutine linkage.
       STX  GWRDRES
       LDX  #GCHRIN
       STX  GCHRRES
       JSR  PWRDIN   Print words until blank line found.
       SWI           Return to operating system.
*
COLINK MACRO FROM,TO Coroutine linkages.
       PULS Y
       STY  FROM
       JMP  [TO]
       ENDM
*
GETCHR  COLINK GWRDRES,GCHRRES
GETWRDG COLINK GCHRRES,GWRDRES
GETWRDP COLINK PWRDRES,GWRDRES
PRTWRD  COLINK GWRDRES,PWRDRES
*
GCHRRES RMB  2       Resumption address for GetChar.
GWRDRES RMB  2       Resumption address for GetWord.
PWRDRES RMB  2       Resumption address for PrintWord.
```

REFERENCES

Recursive algorithms are discussed in *Recursive Programming Techniques* by D. W. Barron [American Elsevier, 1968]. *Programming Language Structures* by Organick, Forsythe, and Plummer [Academic Press, 1978], also contains an extensive discussion of recursion.

Coroutines and their relationship to multipass algorithms are discussed in *Programming Language Structures* and in Knuth's *Fundamental Algorithms* [Addison-Wesley, 1973 (second edition)] The word "coroutine" was coined by M. E. Conway and appears in his paper, "Design of a Separable Transition-Diagram Compiler" [*Comm. ACM*, Vol. 6, No. 7, July 1963, pp. 396-408]. However, Knuth has found the concept mentioned as early as 1954 in a UNIVAC "programming tip."

Many examples of filter programs are given in Kernighan and Plauger's *Software Tools* [Addison-Wesley, 1976]. The idea of cascading filters appears prominently in the UNIX operating system for the PDP-11 and other computers, where such a cascade is called a *pipe*. UNIX's pipes effectively allow a user to link together cooperating programs (coroutines) at run time.

NCC 81

The National Computer Conference (NCC) was held this year at the McCormick Place Convention Center in Chicago. Show dates were May 4-7. While primarily a show for larger manufacturers, this years exhibits included some of our own Standard S50 Bus manufacturers and software vendors. The crowds of domestic and overseas buyers and visitors was large (75,000 so they say) and the entire operations covered all three floors, each about eight acres. The newer generation micros were doing a landslide business, or so you would believe from the reaction I observed. Boy, but my feet sure took a beating.

As detailed in Ron Anderson's column this month (FLEX USER NOTES) and report last month in BIT BUCKET, pg 33, by Jacky Cocklnos of Paris Radio, Australia and the far East, many of our more popular software vendors and engineers were showing their products at the GIMIX double booth. In fear of leaving someone out (as I did in last years report of the Philly show) I will not attempt to name each individual who was there. Instead I will devote the space to pictures and caption them as best as I can remember.

1. GIMIX had the largest booth of all Standard S50 Bus vendors there. In the center right to left is Richard and Arlene Don surrounded by visitors at the GIMIX booths. In addition to the numerous other vendors who were displaying their hard/software at the GIMIX booths, GIMIX was getting a big response to their new 5" winchester disk system and new CMOS ram cards with battery backup. Ken Kaplan of Microware was continually swamped by users and onlookers as he demonstrated Microware's OS9, multi-user, multi-tasking disk system.

Dr. Ray Talbot 'father' of tFORTH was also kept busy with repeated demonstrations of his FORTH series for the 6809. Shelly Epstein of Epstein Associates had some fancy high-res graphics running on a color monitor while Ken Kaplan was demostrating other functions on a GIMIX system running OS9. Both on the same system.

For the entire time that Joyce and I were there the GIMIX booth was continually

crowded with onlookers. I just might admit that I got that smirk, time after time, as I heard show-goers remark, "Boy, the 6800 crowd has really gone to the front", or similar remarks. For four or five years now I have been saying - just you other fellows watch out! It is very satisfying to see the Standard S50 Bus and 68XX vendors doing so well. I put my money where my mouth was, three years ago when despite all the projections of failure by many on 'the other bus' that a 68XX magazine would never make it. Well folks we have - and so has the Standard S50 Bus and all those fine folks who hung in there! I saw many, many systems at this NCC that pale beside the average Standard S50 Bus system. This ole dog don't hang his tail, and the best is yet to come.

2. SMOKE SIGNAL BROADCASTING was another of the fine Standard S50 Bus manufacturers who had a well manned (and girled (?)) booth at the '81 NCC. Did I get it right, Ron?

From right to left is Deborah Conrad, OEM and Dealer executive for SSB and Jim Alday General Sales Manager. Ric Hammond, Smoke president was also there but for two days I kept going back to their booth but Ric was always 'gone off to close a deal' or some other good something or another. SSB was doing a busy pace demonstrating their new 6809 systems and running some impressive applications software. Was told that the 'official' disk system for the Smoke series of computers was to be Microware's OS9. Seems that the multi-users multi-tasking thing is about upon us, and none too soon. The minis will be hard pressed to hold the line in the future.

3. SOUTHWEST TECHNICAL PRODUCTS was represented at the Semi Conductor Specialist's booth at NCC by the folks from MICRO-POWER (Juggler game fame). At the right the guy with a tie and fancy beard is Ed Evans of Micro-Power demonstrating a SWTPC S09 system and the new 8212W Word Processing CRT Terminal. In the background is Paul Yamada, also of Micro-Power. Here we also saw some fine applications running. It seems that there is a lot more business and other software running (tested and proven) that we do not hear much about. This corner of the booth was active every time I went by there and it indicates that there were a LOT of folks wandering around looking for the type systems and software that was running on "OUR" type machines. I heard and saw nothing but good things. I went to the NCC with some doubts, but came away realizing that we are in the thick of things. Despite all the prophets of doom I have had the misfortune to know over the past five or six years, it is apparent that the Standard S50 Bus is well and doing more than just kicking.

4. What? this is no NCC booth, actually what it is is my office away from the office. Complete with 110 volt ac power so I can whip out this stuff anywhere I am. The sweet thing waving is my boss Joyce, for over 36 years now she is what has kept me straight. Here we are parked smack dab in the middle of the Rome, Georgia Airport. Camped here for two days and video taped an airshow. Wondered what all those pilots thought when they were landing and saw us there between runways. Boy, life is sure rough in the wilderness. Couldn't think of any place else to place this picture, so just stuck it in the middle of my ramblings on the NCC 1981.

While on the subject of shows, thought I would mention something now that I hope can take place next year, June 1982.

Each year for the past three or four years we (the whole CPI family - 68 Micro Journal - Data-comp - SouthEast Media) have had a booth at the Atlanta Hamfest. Computers are becoming stronger at this show each year. Of all the shows we attend (which are many) this one we certainly enjoy the most! The crowds are heavy, and the show is short, 2 days a Saturday and Sunday. The response we have received each year has been more than good. Last year alone we

talked to hundreds of 68XX users, at the Atlanta Hamfest. This year we expect to see and get an opportunity to talk to a lot of readers and other old friends.

What I am about to suggest is in my opinion something that we as a group have needed to do for some time now.

We need a 68XX mini-convention!

Atlanta is an ideal place to hold such a meeting and those vendors and manufacturers of Standard S50 Bus wares, that I have discussed this with so far, feel that it could be a good experience for all of us. In fact we just might want to do it every year, its up to you.

The hotel rates and the cooperation of the management there has been the most cooperative that I have seen anywhere. There are many fine resturants and sightseeing places very close by and the show goers are the finest of any area. Fact is the crowd is drawn from all over for the hamfest. Booth rates are very reasonable (best yet) and no monkey business about who hauls your stuff in or plugs in your extension cord. Northern shows are fine, but the difference in prices and many other considerations makes Atlanta a natural.

Have not discussed this with the folks who run the hamfest (could also be called computerfest) but if we can not do it at their thing we could always do it by ourselves. My feelings are, and remember it has not been discussed, that we would be welcome. They are just that kind of folks. Surprising how many radio operators we meet that have a hankering for computing. Also last year gained our largest OEM account at this show.

Well, give it some thought and then let me know. If enough of you evidence some interest I am sure that we can get many of the folks who show at other shows to come and let us all see their new goodies. But when it is all said and done the best part for me, is getting to meet so many of you. I don't mind admitting at all, that when I think back over the past five or six years, I realize that some of the finest folks I have ever known are you, our readers and loyal supporters. Sometimes as I talk to some of you, or read your letters to us, I get the feeling that we are more a fraternal club than a computer magazine, and that just suits me fine. Thanks!

DMW - - - -

NEED $$?

More and more I am getting calls from advertisers and others who are entering the 6809 market, who need some part time, off site help. Many are pressed with the need to have some immediate software projects completed. Right now I have a hardware manufacturer who needs someone to write some drivers, for one of the more popular 6809 disk operating systems. An excellent opportunity for someone to pick up a nice piece of quick income.

Most of these projects can pay you well for a weekend or two of code engineering. Fact is I know of many who started out doing this sort of thing, and now work full time at it, for better than average programmers wages. And best of all they never leave home to go to work.

The primary points are that you develope 'GOOD' code and that you get the job finished on time. Of course you will have to have your own 6809 system to work on. Also you should expect to maintain the code, or at least take prompt care of any 'bugs' that might have slipped in. Your reputation will spread, good or bad, and you will be surprised at the money to be made.

I have put about everyone I personally know to work, still the requests come in for contract programmers. Also occasionally hardware types, but the main needs seem to be code engineers.

If you are interested send me a short letter and tell me what you have done and what you think you can do. The money is between you and the other party so I don't need to get into that. Let me know what your system consist of and what software fields you believe that you can do a professional job in. If I think you are the person I will pass your name, address and telephone number along and you may be getting a call. From there on you are on your own!

If you are programming for some other company that might be offended if you sold your services elsewhere, then please DON'T get in on this because I feel that we have to consider loyalty foremost. But if you can, without conflict, devote some spare time toward earning some additional income, please let me know. If you are not a professional but a 'darned good hobbyist' you just might be able to fill the bill (I know many who have), so you let me know also.

DMW - - - -

COLOR

For the past few months, as our TRS80 Color Computer readers grow in number, we are receiving more and more items for the color computer, to review.

Therefore, we need those readers, who have the Radio Shack color computer, with 16K and extended BASIC, who would like to become a 'reviewer', to drop me a line. Tell me what you have (computers), some background on your experience and most important, how fast you think it will take you to get an honest, impartial and accurate review of the product. Remember, it is no problem to get someone who will review the documentation, we get too many of those kind (which we do not publish). What I want is a COMPLETE review of the total package, nothing less!

In most cases your reward will be to KEEP the product reviewed. This is our standard policy for reviews.

Some items I want a novice to review. Especially if it is advertised 'as easy' to use or build. Other items I want more professional and technical input, but in either case I can use some of you.

It has been the norm, in the past, that once I gain confidence in a reviewer, I send them additional items to review. So if you do a bang up job, promptly, you will reap all kinds of rewards, if the products keep coming in. If you are sloppy, and cannot turn out a review that I can photo-reduce or copy, then it will not work. Please remember, it is VERY IMPORTANT to get a review back promptly and above all accurate. The folks who sent the item for review need it published AS SOON AS WE CAN! Reviews sell more products and help the newcomers off to a much better start.

So if you are interested, let me know - soon.

DMW - - - -

Dale L. Puckett
14753 Endsley Turn, Woodbridge, Va. 22193
Don Williams, Editor '68 Micro Journal
Dear Don,

During the past several months we have made several improvements to the 6809 version of READTEST. I hope you will be able to pass this information to your many readers.

READTEST is now compatible with STYLOGRAPH text files. It recognizes the comma as a text processor command as well as the period and the colon used by the TSC text processor.

READTEST now reports the percentage of personal words and the number of affixes per 100. These figures were used before in the computation of overall readability, but were not printed. They give the writer more information to work with when evaluating his writing.

READTEST now contains a routine that checks to see if the user has left his FLEX width parameter set to zero. If he has, READTEST now defaults to a terminal width of 64 characters. If the user has set his terminal width in FLEX, READTEST uses it.

An overflow condition that existed when evaluating a text file that was very difficult to read because it contained more than 650 affixes, ie, 1200 words with more than 57 affixes per 100 words, has been eliminated.

A typo that slipped into the final source code and caused READTEST to print the last part of its report twice every time a "DIFFICULT" message was required has been eliminated. By the way, this error never appeared in the 6800 code.

The 6809 version of READTEST now contains an improved word counting routine that completely eliminates any ambiguity when confronted with a file containing a mixture of straight text and text processor commands.

Finally, I would like to pass along one FANTASTIC idea forwarded to us by a READTEST owner, Martin J. Petersen, Jr. of the Harris Corporation Broadcast Products Division in Quincy, Ill. Mr. Petersen prints a copy of READTEST's report and submits it with his manuscripts when he offers them for publication. He believes that this shows the editor that he cares about his writing. I believe he is right.

Thanks for sharing this information with your readers.

Best Regards,

Dale

Support Our ADVERTISERS!

SURVEY

I have recently completed a survey of a selected group of 68XX users. Not a strictly formal, but a very enlightening survey. It was conducted mainly by telephone, with a sampling from correspondence and notes on subscription renewals (the ones that ask - what kind of computer, etc.). The number of users sampled gives me a better plus or minus factor than most other polls. Even though some aspects date back two years or more, the ratio remains practically the same.

The data I was most interested in fell into the following categories:

Did they feel that the 6800 is obselete? Did they think that if they could still purchase a good, fast low price 6800 system (2 mhz, 64K memory, disk, serial and parallel interfaces) would they be interested? Did they believe that a disk system was a 'must'? Of the following which is the most important-price( ), quality( ), support( ), service( ), brandname( ), in order check 1 thru 5? Is the system used for hobby( )-business( ), check one or both? What make (manufacturer's name) of computer? What processor 6800-6809-68000? What type data storage system (tape - disk)? If disk, what size and how many? If tape what type interface? What is the primary and secondary use of the system? What operating systems (tape - disk) was used? What software (designer - application) purchased? What software (designer - application) used? What software (designer - application) available but not used? Did they have any software that was a commercial item but they had not purchased? Concerning hardware and software did they feel strongly concerning the portability of software between different systems important? Would they refuse to purchase hardware or software that was not completely compatible with existing hardware and popular disk systems? If so why? What they liked most about their hardware? What they liked least about their hardware? If they are going to purchase more hardware, in the future, then who from (dealer, make, type). What they liked best about their software? What they liked least about their software? If they are going to purchase more software, in the future, then who from (dealer, make, type)? Of all the companies dealt with on hardware, who gave the best support? Why? Of all the companies dealt with concerning hardware how would they rate them on support, best thru worst? Of all the companies dealt with on software, who gave the best support? Why? Of all the companies dealt with concerning software how would they rate them on support, best thru worst? Considering the present supply of hardware and software available, what would they like to have made available, that is not now, in the hardware and software line? Last but very important, what did they think of the future of computers as we now know them (bus, processor and storage devices)? The last of course very general but essential to form a 'gut' feeling of overall user opinion.

Now, I have some fairly accurate figures for the above survey, but I need to give all of you a chance to tell me your answers to the above also. So here is what I propose to do: If you will fill in the survey on the reverse and send it back to me, here at 68 Micro Journal, I will extend your subscription for one (1) month. I know this is not a lot but it just may help you in some way in the future. I know for a fact that most all those who advertise and sell the computer things we all buy will be very interested in what you report. All replies must be received back by September 15, 1981 to qualify for the subscription extension.

DMW - - - -

6800-6809-68000-ETC. SURVEY 1981

IS THE 6800 OBSOLETE YES( )-NO( )? IS A DISK SYSTEM A MUST TO 'YOU' YES( )-NO( )? DOES THE 68000 INTEREST YOU YES( )-NO( )?

IF YOU COULD PURCHASE A FAST 6800 SYSTEM, FULL MEMORY, WITH DISK, SERIAL AND PARALLEL INTERFACES- WOULD YOU BE INTERESTED YES( )-NO( )?

WHICH IS THE MOST IMPORTANT-PRICE( ), QUALITY( ), SUPPORT( ),SERVICE( ), BRANDNAME( ), IN ORDER FILL-IN RATING 1 THRU 5?

YOUR SYSTEM USE-BUSINESS( ),HOBBY( )-CHECK ONE OR BOTH.

SYSTEM BRANDNAME _____

WHAT CPU 6800( ),6809( ),68000( ),OTHER( )? WHAT TYPE STORAGE-TAPE( ),DISK( ).

IF DISK WHAT SIZE AND MAKE_____ IF TAPE WHAT BRAND INTERFACE_____

WHAT OPERATING SYSTEM-TAPE OR DISK_____ WHAT MONITOR_____

WHAT SOFTWARE PURCHASED_____

WHAT SOFTWARE USED_____

DO YOU HAVE ANY COMMERCIAL SOFTWARE THAT WAS NOT PURCHASED BY YOU-YES( ),NO( )?

DO YOU FEEL THAT SOFTWARE AND HARDWARE MUST BE COMPATIBLE BETWEEN DIFFERENT SYSTEMS AND MANUFACTURE-YES( ), NO( )?

WHO GAVE YOU THE BEST SUPPORT-SOFTWARE-HARDWARE_____

_____

IF YOUR ARE GOING TO PURCHASE HARDWARE IN THE FUTURE (DEALER,MAKE,TYPE)_____

_____

IF YOU ARE GOING TO PURCHASE SOFTWARE IN THE FUTURE (DEALER,TYPE)_____

_____

WHAT DO YOU LIKE BEST ABOUT YOUR SOFTWARE_____

_____LEAST_____

WHAT DO YOU LIKE BEST ABOUT YOUR HARDWARE_____

_____LEAST_____

OF ALL COMPANIES DEALT WITH WHO WAS BEST_____

WHY?_____

OF ALL COMPANIES DEALT WITH WHO WAS WORST_____

WHY?_____

_____

OF ALL COMPANIES DEALT WITH-RATE THEM ALL-1 THRU X-(1=BEST, X=WORST)_____

_____

_____

WHAT NEW PRODUCTS WOULD YOU LIKE TO HAVE AVAILABLE_____

_____

_____

DO YOU HAVE OTHER COMPUTER SYSTEMS? IF SO (NAME-CPU-ETC,)_____

ALL SPACES MUST BE COMPLETED TO QUALIFY FOR SUBSCRIPTION EXTENSION! ALSO THEY MUST BE RETURNED TO 68 MICRO JOURNAL BY SEPTEMBER 15, 1981.

PLEASE USE AN ADDITIONAL SHEET OF PAPER AND STATE WHAT YOU THINK OF THE FUTURE OF OUR COMPUTERS, BUS, PROCESSOR, STORAGE DEVICES, ETC.

MAKE SENSE ?

HOW TO WRITE DOCUMENTATION
John P. Tucker
POB 2898
Laredo, TX 78041

Or Do As I Say, Not As I Do!

There are those of us, and we number in the thousands, who hunger for good software. There are those of you, and you number in the dozens, who feed us well.

And then there are those, both of you, who document your software so that we thousands know what is going on.

In the past two months I have received four excellent software programs. Two of them are still in the desk drawer. I'm trying to figure out what they are supposed to do. The samples included on the disks run well; the documentation hardly gives a clue as to the purpose of the programs. The construction of files to run within the programs is still a mystery.

A third set of programs offers me a superbly unique way of managing data files — if I ever figure out what steps to take in what order, how to create and arrange the data files in the beginning, and how to get them "circulating" within the programs. Again, the samples run beautifully, but the documentation is too sparse.

All three programs even furnish listings of the source code. Were I a Source Code Reader, perhaps the mystery would be solved. But I am a Documentation Reader.

That is why the fourth program is such a joy. No source code came with the program. It cannot even be disassembled with an ASCII disassembler. It seems to be written partly in ASCII, partly in binary, partly in decimal, and partly in Spencerian Greek. Yet I can use the progre. with no hesitation. The documentation leaves no questions unanswered, gives hints and pointers where required, and when I do something unforgiveable, it tells me so in English, politely and concisely and informatively. Since this is one of the most complex programs available, the documentation must be written correctly. The program is TSC's Extended Precision Basic.

Who was that screaming, "But that's different! That's TSC!"?

Dadgum it all, it is NOT different! You can, and should — repeat, should — write documentation equal to TSC's and Heathkit's. You need to do so even more than those two companies! You don't have the time or facilities to answer interminable streams of questions. Suppose you have written the JET DATA DEVELOPMENT AND MANAGEMENT PROCEDURES and offered them for sale. I would certainly hope that the documentation would run something like this:

The JET DD&M PROCEDURES are intended as an aid and system in developing files of names, addresses, and personal data suitable for creating mailing lists, geneological research files, personnel records and directories for companies and civic organizations, and even small inventory records. By prefixing each entry with what is known as a Key Symbol, these files can then be sorted using any Key or any group of Keys to list, omit, or arrange the output as desired. Typical Key Symbols used as prefixes are <further data and information>.

STEP NUMBER ONE: Configure the program to your computer and terminal. This is done by booting up Flex 2.0, assigning your System Disk as #0 and your Working Disk as #1 (S=0,W=1).

LEAVE NOTHING TO THE IMAGINATION

You then proceed to tell in complete (COMPLETE) detail each change that is required. If you are trying to set up cursor controls, tell the reader "On my system, using an XZ-7121 terminal, to move the cursor down three lines the command is ESCAPE;C. To move it to the right six spaces the command is ESCAPE;M,6" or whatever is correct. Then tell the reader that "You change these commands by <calling up the menu and selecting CHANGE>,<altering the Basic program at lines xxxx to yyyy>, or whatever method is used in your program. Give details of what to change, where to find the information to be changed, and how many changes total will be required!

And provide a simple little sample program the user can run to test his changes!

What's next? Does your program require some dummy files to be on disk before it can start running? Say so! And in many words! "Before the first operations can be attempted, the following dummy files must be placed on the Working Disk. A file named <DUMMY><specify extension> must be created containg the following entries (give the entries EXACTLY); a second file...etc.

Where do we go from here? Do we create text files? Tell us, "The next step is to create the Text Files you will be using. The very first entry on any Text File for this system must be (whatever it must be).

AGAIN, ASSUME NOTHING!

Don't even assume I know how to save a text file under your program. Remember, you worked with that program, you debugged it (I hope) until it runs like a fine watch...at least, a good Timex, like I wear. You know that program. You know each quirk and entry mode. But, I don't know anything about your program except that it is on a disk and came with a sheet of paper that said, "Load this program under Flex 2.0 and run it." At least you told me it took Flex 2.0 — I have one on hand that must be written in Jinx 7.25.

Or Pascal, which is worse.

If you really want to learn to write documentation, buy a small Heathkit for a device that you know absolutely nothing about. Read everything Heath sends you.

There will even be a sheet on how to use a soldering iron! Build the device, following the instructions to the letter. It will work.

Now, sit down and write me documentation for your program. Each time you make an entry on your keyboard, write down what entry to make and why.

IMPORTANT! All of this information must be placed in a single section, totally divorced from your explanations of each segment of your program. Number the pages. Let me start at Page 1, doing little exercises as I go along, and wind up at Page X at least with a comprehension of what I am trying to accomplish. Then perhaps, by re-reading from Page 1, I can refer to the various sections of your well-written program and know how to use it to its fullest advantage.

That separate section, standing alone, is a MUST in well-written program documentation. (See the mini-tutorial for the TSC Text Editor. It has you editing text and knowing why before you are even introduced to the real power of the system. The power comes easily after that lesson.)

Do you want to sell twenty copies of your program to twenty experienced programmers, or do you want two thousand copies in the hands of that many happy users?

You alone have the answer. Your documentation makes either one possible.

CONCLUSION: Don't never assume that nobody knows nothing about whatever.

DISKSAVE

DISKSAVE UTILITY JOHN CHAMPLAIN

THIS PROGRAM IS AN EXPANDED VERSION OF 'SECTOR',WHICH WAS WRITTEN BY BILL KNIGHT,AND PUBLISHED IN '68' MICRO ,JUNE 1980. THIS PROGRAM WILL ALLOW YOU TO RECOVER DATA AND FILES FROM A DISK THAT HAS LOST ITS DIRECTORY. I HAVE BEEN ABLE TO RECOVER A COMPLETE DISK SUCSESSFULLY BY USING THIS ROUTINE. IT DOES REQUIRE THAT YOU HAVE SOME KNOWLEDGE OF HOW FLEX DISKS ARE FORMATTED. THIS ARTICLE IS NOT INTENDED TO BE A TUTORIAL ON FLEX. IF YOU HAVE A COPY OF THE FLEX PROGRAMMER'S MANUAL IT WILL HELP YOU UNDERSTAND HOW THIS ROUTINE RECOVERS FILES.

THIS PROGRAM IS CALLED FROM FLEX BY TYPING 'DISKSAVE'. NO OTHER PARAMETERS ARE NECESSARY. AFTER THE COMMAND TABLE IS DISPLAYED ON YOUR SCREEN,INSERT THE BAD DISK IN DRIVE #1 AND A GOOD DISK IN DRIVE #0. IT MIGHT BE WISE TO USE A BLANK DISK IN DRIVE 0 BUT IS NOT REALLY NECESSARY,JUST BE SURE THE DISK YOU USE HAS ENOUGH FREE SPACE LEFT ON IT.

THE COMMANDS ARE:

R - READ SECTORS THIS COMMAND UPON ENTRY WILL PROMPT YOU FOR TRACK AND SECTOR ADDRESS INPUT. IT WILL THEN ASK FOR A DRIVE #. ANSWERS MUST BE A TWO DIGIT HEX NUMBER FOLLOWED BY A CR. YOU CAN LOOK AT THE DIRECTORY IF YOU WISH,BUT BE AWARE THAT THE SECTOR MAY NOT BE READABLE AND THE DISPLAY MAY NOT BE CORRECT. I HAVE NOT YET FIGURED OUT HOW TO GET AROUND THIS PROBLEM. IF YOU HAVE A SOLUTION, PLEASE LET THE REST OF US KNOW. YOU MUST ISSUE THE 'R' COMMAND BEFORE ANY OF THE OTHER COMMANDS (EXCEPT 'M') WILL RESPOND. AS EACH SECTOR IS READ, ITS LOCATION ON THE DISK IS ADDED TO A MAP TABLE FOR USE BY THE OTHER ROUTINES. THIS TABLE IS RESET EACH TIME THE 'R' COMMAND IS ISSUED.

N - READ NEXT SECTOR READS THE NEXT LOGICAL SECTOR IN THE CHAIN THAT YOU STARTED WITH AND ENTERS ITS LOCATION IN THE MAP TABLE. B - BACKUP READS AND DISPLAYS THE LAST SECTOR READ BEFORE THE CURRENT ONE. IT ALSO RESETS THE MAP TABLE.

T - DISPLAY SECTOR MAP TABLE THIS COMMAND READS THE MAP TABLE AND DISPLAYS EACH TRACK AND SECTOR READ .

D - DISPLAY COMMANDS

M - RETURN TO MONITOR MAY BE ISSUED WHENEVER THE PROGRAM ASKS FOR A COMMAND AND RETURNS TO FLEX.

S - SAVE TO NEW DISK UPON SELECTION YOU WILL BE ASKED TO TYPE IN A FILE NAME. UP TO 8 LETTERS CAN BE USED. YOU MAY ALSO SPECIFY AN EXTENSION OF YOUR CHOICE (3 LETTERS). DEFAULT EXTENSION IS '.BAC'. A CR WILL CAUSE THE SECTORS IN THE MAP TABLE TO BE WRITTEN TO THE NEW DISK ON DRIVE 0. FLEX WILL AUTOMATICALLY ASSIGN NEW TRACK AND SECTOR LOCATIONS AS EACH SECTOR IS BEING WRITTEN. WHEN YOU ARE FINISHED, YOU CAN THEN USE ANY OTHER APPROPRIATE UTILITYS YOU HAVE ON THE RECOVERED FILE.

C - CHANGE DATA THIS ROUTINE ALLOWS YOU TO CHANGE ANY DATA BYTE THAT YOU SELECT. PLEASE USE CAUTION AS YOU COULD CAUSE DATA TO BE LOST.

I WOULD APPRECIATE HEARING FROM YOU IF YOU EXPAND OR IMPROVE THIS PROGRAM. BETTER YET, WHY NOT SEND YOUR CHANGES TO '68' MICRO SO ALL CAN SHARE.

INTEXT

INTEXT UTILITY

The INTEXT utility is a text entry utility written for the 6809 operating under FLEX™. In order to use the utility your entry terminal must be capable of sending each character as it is entered and must support $08 as the backspace character. INTEXT may easily be rewritten for the 6800.

INTEXT accepts character entries and begins storing them in a buffer in main user memory at $0200. The buffer ends (as written) at $9FFF which gives you a text capacity of nearly 41,000 characters. When you are within 256 characters of the end of the buffer, you are given a warning and can then write your text to disk. When the INTEXT command is called it first prompts for the maximum line length of lines in the output file. It then gives you a table of abbreviated commands which may be inserted into your text. These abbreviated commands represent some commands recognized by TSC's Text Processor (.SP, .SI5, .UL). The abbreviated commands will be expanded in the output file and arranged for direct use by the Text Processor.

You can now begin typing your text. Just type away and do not worry about entering any carriage returns. When you have completed entering your text, enter a CTRL C. INTEXT will then prompt for the output file name (default extension is .TXT). INTEXT then begins processing the buffer. It counts characters until it reaches the line length limit. If that character happens to be a space, then INTEXT substitutes a carriage return and continues with the next line. If the character is not a space, INTEXT backs up until it finds the preceeding space and substitutes the carriage return at that location. It then continues processing text from that point.

You may find that you need to "clean up" the first line of the output file as it may have an undesired carriage return. This happens if you begin with one of the text processor commands. The text processor commands are as follows:

^=.SP

~=.SI5

UNDERLINED= UNDERLINED

The Text Processor commands may be inserted alone or in any combination. The INTEXT utility makes entry of bulk text very simple. You can spend your time looking at the copy you are typing and not worry about when to type a carriage return.

```
TEXT INPUT PROGRAM                          5-24-81 TSC ASSEMBLER   PAGE    1
WITH AUTO CARRIAGE RETURN


                        * PROGRAM FILENAME: INTEXT.CMD

                        * All Rights Waived

                        * John C. Tarvin
                        * 14680 Shadowlane Court
                        * Morgan Hill, CA  95037
                        * (408) 683-0287

                        * PROGRAM AND DOS EQUATES

                CD4B    INDEC   EQU     $CD4B       INPUT DECIMAL NUMBER
                CD1B    INBUFF  EQU     $CD1B       INPUT LINE TO BUFFER
                0406    PMB     EQU     $0406
                CD03    WARMS   EQU     $CD03       FLEX WARM START
                CD15    GETCHR  EQU     $CD15       GET CHAR. FROM TERMINAL
                CD1E    PSTRNG  EQU     $CD1E       OUTPUT A STRING
                CD33    GETEXT  EQU     $CD33       GET DEFAULT EXTENSION
                CD3F    RPTERR  EQU     $CD3F       REPORT ERROR
                CD2D    GETFIL  EQU     $CD2D       GET FILE SPEC

C100                            ORG     $C100

                        * CONSTANTS AND RESERVATIONS

C100 CD4B               FCB     FDB     $CD4B
C102 0200               BUFF1B  FDB     $0200       INPUT BUFFER BEGINNING
C104 0000               BUFF2B  FDB     $0000       OUTPUT BUFFER BEGINNING
C106 9FFF               BUFF1E  FDB     $9FFF       INPUT BUFFER LIMIT
C108                    BUFF1P  RMB     2           BUFFER POINTER
C10A                    LINLEN  RMB     1           LINE LENGTH
C10B 08                 BKSPC   FCB     $08         BACKSPACE CHAR. DEFINITION
C10C 0D                 CR      FCB     $0D         BR CHAR. DEFINITION
C10D 66                 MAXLIN  FCB     102         MAX LINE LENGTH
C10E 03                 ETX     FCB     $03         END OF SESSION
C10F 9F00               BUFWRN  FDB     $9F00       BUFFER WARNING POINT

C111 30  8D 0192        START   LEAX    MSG1,PCR    GET LINE LENGTH
C115 BD  C01E                   JSR     PSTRNG
C118 BD  CD1B                   JSR     INBUFF
C11B BD  CD4B                   JSR     INDEC
C11E 1F  10                     TFR     X,D         PUT NUMBER INTO D
C120 E1  8C EA                  CMPB    MAXLIN,PCR  TOO LONG?
C123 2F  09                     BLE     START1
C125 30  8D 01 E                LEAX    MSG2,PCR    LENGTH ERROR MSG
C129 BD  CD1E                   JSR     PSTRNG
C12C 20  E2                     BRA     START
C12E F7  C10A           START1  STB     LINLEN      SAVE LINE LENGTH
C131 30  8D 0180                LEAX    MSG3,PCR    INSTRUCTIONS
C135 BD  CD1E                   JSR     PSTRNG

                        * TYPE INTO INPUT BUFFER

C138 10AE 8C C6         TYPE    LDY     BUFF1B,PCR  POINT TO INPUT BUFFER
C13C BD  CD15           TYPE1   JSR     GETCHR      GET A CHARACTER
C13F A1  8C CC                  CMPA    E1+,PCR     THROUGH TYPING
C142 27  30                     BEQ     ALIGN
C144 A1  8C C4                  CMPA    BKSPC,PCR   BACKSPACE?
C147 27  15                     BEQ     TYPE2
C149 A1  8C C0                  CMPA    CR,PCR      CR?
C14C 27  EE                     BEQ     TYPE1
C14E A7  A0                     STA     ,Y+         SAVE CHAR. AND INCREMENT
C150 10AC 8C B2                 CMPY    BUFF1E,PCR  BUFFER FULL?
C154 27  0C                     BEQ     STOP
C156 10AC 8C B5                 CMPY    BUFWRN,PCR  WARNING POINT?
C15A 27  0F                     BEQ     WARN
C15C 20  DE                     BRA     TYPE1
C15E 31  3F             TYPE2   LEAY    -1,Y        BACK UP ONE
C160 20  DA                     BRA     TYPE1       GET NEXT CHAR
C162 30  8D 01F1        STOP    LEAX    MSG4,PCR    BUFFER LIMIT REACHED
C166 BD  CD1E                   JSR     PSTRNG
C169 20  09                     BRA     ALIGN
C16B 30  8D 0224        WARN    LEAX    MSG5,PCR    WARNING MSG
C16F BD  CD1E                   JSR     PSTRNG
C172 20  C9                     BRA     TYPE1

                        * GET FILENAME

C174 9DBF C140          ALIGN   STY     BUFF1P
C178 30  8D 0228                LEAX    MSG6,PCR    GET FILENAME MSG
C17C BD  CD1E                   JSR     PSTRNG
C17F BD  CD1B                   JSR     INBUFF
C182 AE  8D FF7A        *       LDX     FCB,PCR     POINT TO FCB
C186 BD  CD2D                   JSR     GETFIL      PUT SPEC IN FCB
C189 1025 0115                  LBCS    ERROR
C18D 86  01                     LDA     #1          SET DEFAULT EXT TO TEXT
C18F BD  CD33                   JSR     GETEXT

                        * RESTRUCTURE INPUT TO ADJUST LINE LENGTH

C192 10AE 8D FF68               LDY     BUFF1B,PCR  Y INT TO INPUT BUFFER START
C197 AE  8D FF69                LDX     BUFF2B,PCR  POINT TO OUTPUT BUFF START
C19B E6  8D FF68        ALIGN3  LDB     LINLEN,PCR  LOAD LENGTH COUNTER
C19F A6  A0             ALIGN1  LDA     ,Y+         GET CHAR FROM INPUT
C1A1 81  5E                     CMPA    #$5E        IS IT ^ REQUESTING
C1A3 27  38                     BEQ     SPACE       .SP INSERTION?
C1A5 81  7E                     CMPA    #$7E        IS IT ~ REQUESTING
C1A7 27  57                     BEQ     INDENT      SINGLE INDENT?
C1A9 81  5F                     CMPA    #$5F        IS IT A _ REQUESTING
C1AB 27  7B                     BEQ     UNDLIN      UNDERLINE
C1AD A7  80                     STA     ,X+         PUT IN OUTPUT BUFFER

C1AF 10AC 8D FF34               CMPY    BUFF1P,PCR  END OF BUFFER?
C1B4 1027 00A9                  LBEQ    SAVEIT
C1B8 5A                         DECB                LOOP TILL THROUGH
C1B9 26  E4                     BNE     ALIGN1
C1BB 31  3F                     LEAY    -1,Y
C1BD 30  1F                     LEAX    -1,X
C1BF 81  20                     CMPA    #$20        WAS CHAR A SPACE?
C1C1 27  10                     BEQ     REPLC
C1C3 31  3F                     LEAY    -1,Y
C1C5 30  1F                     LEAX    -1,X
C1C7 A6  A4             ALIGN2  LDA     ,Y          GET CHAR
C1C9 81  20                     CMPA    #$20        WAS IT A SPACE?
C1CB 27  06                     BEQ     REPLC
C1CD 31  3F                     LEAY    -1,Y
C1CF 30  1F                     LEAX    -1,X
C1D1 20  F4                     BRA     ALIGN2
C1D3 A6  8D FF35        REPLC   LDA     CR,PCR      GET CR
C1D7 A7  80                     STA     ,X+
C1D9 31  21                     LEAY    1,Y
C1DB 20  BE                     BRA     ALIGN3
C1DD A6  1F             SPACE   LDA     -1,X        GET PREV. CHAR.
C1DF 81  0D                     CMPA    #$0D        IS IT A CR?
C1E1 27  04                     BEQ     SPACE1
```

```
C1E3 86    0D              LDA    #$0D        LOAD CR
C1E5 A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C1E7 86    2E      SPACE1  LDA    #$2E        LOAD A PERIOD
C1E9 A7    80              STA    ,X+         PUT IN BUFFER
C1EB 86    53              LDA    #'S         LOAD AN S
C1ED A7    80              STA    ,X+         PUT IN BUFFER
C1EF 86    50              LDA    #'P         LOAD A P
C1F1 A7    80              STA    ,X+         PUT IN BUFFER
C1F3 86    0D              LDA    #$0D        LOAD A CR
C1F5 A7    80              STA    ,X+         PUT IN BUFFER
C1F7 10AC  8D FF0C         CMPY   BUFFIP,PCR END OF INPUT?
C1FC 27    A3              BEQ    SAVEIT      IF SO, WRITE TO DISK
C1FE 20    98              BRA    ALIGN3      GET MORE TEXT
C200 A6    1F      INDENT  LDA    -1,X        GET PREV. CHAR.
C202 81    0D              CMPA   #$0D        IS IT A CR?
C204 27    04              BEQ    INDEN1
C206 86    0D              LDA    #$0D        LOAD CR
C208 A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C20A 86    2E      INDEN1  LDA    #$2E        LOAD A PERIOD
C20C A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C20E 86    53              LDA    #'S         LOAD AN S
C210 A7    80              STA    ,X+         PUT IN BUFFER
C212 86    49              LDA    #'I         LOA AN I
C214 A7    80              STA    ,X+         PUT IN BUFFER
C216 86    35              LDA    #$35        LOAD A 5
C218 A7    80              STA    ,X+         PUT IN BUFFER
C21A 86    0D              LDA    #$0D        LOA A CR
C21C A7    80              STA    ,X+         PUT IN BUFFER
C21E 10AC  8D FEE5         CMPY   BUFFIP,PCR CHECK END OF BUFFER
C223 27    3C              BEQ    SAVEIT
C225 16    FF72            LBRA   ALIGN3
C228 A6    1F      UNDLIN  LDA    -1,X        GET PREV. CHAR.
C22A 81    0D              CMPA   #$0D        IS IT A CR?
C22C 27    04              BEQ    UNDLN1
 225 86     D              LDA    #$0D        LOAD CR
C230 A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C232 86    2E      UNDLN1  LDA    #$2E        LOAD A PERIOD
C234 A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C236 86    55              LDA    #'U         LOAD A U
C238 A7    80              STA    ,X+         PUT IN BUFFER
C23A 86    4C              LDA    #'L         LOAD AN L
C23C A7    80              STA    ,X+         PUT IN BUFFER
C23E 86    0D              LDA    #$0D        LOAD A CR
C240 A7    80              STA    ,X+         PUT IN BUFFER
C242 10AC  8D FEC1         CMPY   BUFFIP,PCR END OF INPUT BUFF?
C247 27    18              BEQ    SAVEIT
C249 A6    A0      UNDLN2  LDA    ,Y+         GET CHARACTER
C24B 81    5F              CMPA   #$5F        IS IT AN UNDERLINE?
C24D 27    0B              BEQ    UNDLN3
C24F A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C251 10AC  8D FEB2         CMPY   BUFFIP,PCR END OF INPUT BUFF?
C256 27    09              BEQ    SAVEIT
C258 20    EF              BRA    UNDLN2
C25A 86    0D      UNDLN3  LDA    #$0D        LOAD A CR
C25C A7    80              STA    ,X+
C25E 16    FF?A            LBRA   ALIGN3

                   * WRITE FILE TO DISK

C261 86    0D      SAVEIT  LDA    #$0D        LOAD A CR
C263 A7    80              STA    ,X+         PUT IN OUTPUT BUFFER
C265 BF    C108            STX    BUFFIP      SAVE NEW POINTER
C268 AE    8D FE94         LDX    FCB,PCR     POINT TO FCB
C26C 86    02              LDA    #2          WRITE CODE
C26E A7    84              STA    ,X          PUT IN FCB
C270 BD    D406            JSR    FMS         CALL FMS
C273 26    2D              BNE    ERROR       ERROR?
C275 10AE  8D FEBA WRITE   LDY    BUFF2B,PCR POINT TO BUFFER
C27A AE    8D FEB2         LDX    FCB,PCR     POINT TO FCB
C27E 86    00              LDA    00          GET WRITE CODE
C280 A7    84              STA    ,X          PUT IN FCB
C282 A6    A0      WRITE1  LDA    ,Y+         GET CHARACTER
C284 AE    8D FE7D         LDX    FCB,PCR     POINT TO FCB
C288 BD    D406            JSR    FMS         CALL FMS
C28B 26    15              BNE    ERROR       ERROR?
C28D 10AC  8D FE76         CMPY   BUFFIP,PCR THROUGH?
C292 26    EE              BNE    WRITE1
C294 AE    8D FE6B         LDX    FCB,PCR     POINT TO FCB
C298 86    04              LDA    #4          CLOSE CODE
C29A BD    D406            JSR    FMS         CALL FMS
C29D 26    03              BNE    ERROR       ERROR?
C29F 7E    CD03            JMP    WARMS

                   * ERROR ROUTINE

C2A2 BD    CD3F    ERROR   JSR    RPTERR
C2A5 7E    CD03            JMP    WARMS

                   * MESSAGES

C2A8 45 4E 54 45   MSG1    FCC    'ENTER LINE LENGTH (MAX. 102): '
C2C6 04                    FCB    $04
C2C7 4C 49 4E 45   MSG2    FCC    'LINE LENGTH GREATER THAN 102!'
C2E4 04                    FCB    $04
C2E5 54 59 50 45   MSG3    FCC    'TYPE <CTRL C> TO END SESSION'
C301 0D 0A                 FCB    $0D,$04
C303 54 59 50 45           FCC    'TYPE ^ TO ENTER A .SP'
C318 0D 0A                 FCB    $0D,$0A
C31A 54 59 50 45           FCC    'TYPE ^ TO ENTER A .SI5'
C331 0D 0A                 FCB    $0D,$0A
C333 54 59 50 45           FCC    'TYPE _WORD OR WORDS_ TO UNDERLINE'
C354 0D 0A 04              FCB    $0D,$0A,$04
C357 42 55 46 46   MSG4    FCC    'BUFFER FULL, MUST OUTPUT TO DISK!'
C378 04                    FCB    $04
C379 07            MSG5    FCB    $07
C37A 57 41 52 4E           FCC    'WARNING: WITHIN 256 CHARACTER OF LIMI
C3A0 0D 0A 0A 04           FCB    $0D,$0A,$0A,$04
C3A4 45 4E 54 45   MSG6    FCC    'ENTER FILENAME: '
C3B4 04                    FCB    $04

                   * END

                           END    START

0 ERROR(S) DETECTED
```

TEXT INPUT PROGRAM
WITH AUTO CARRIAGE RETURN

SYMBOL TABLE:

ALIGN	C174	ALIGN1	C19F	ALIGN2	C1C7	ALIGN3	C198	BKSPC	C108
BUFF1B	C102	BUFF1E	C106	BUFF1P	C108	BUFF2B	C104	BUFWRN	C10F
CR	C10C	ERROR	C2A2	ETX	C10E	FCB	C100	FMS	D406
GETCHR	CD15	GETFIL	CD2D	INBUFF	CD1B	INDEC	CD48	INDEN1	C20A
INDENT	C200	LINLEN	C10A	MAXLIN	C10D	MSG1	C2A8	MSG2	C2C7
MSG3	C2E5	MSG4	C357	MSG5	C379	MSG6	C3A4	PSTRNG	CD1E
REPLC	C1D3	RPTERR	CD3F	SAVEIT	C261	SETEXT	CD33	SPACE	C1DD
SPACE1	C1E7	START	C111	START1	C12E	STOP	C102	TYPE	C138
TYPE1	C13C	TYPE2	C15E	UNDLIN	C228	UNDLN1	C232	UNDLN2	C249
UNDLN3	C25A	WARMS	CD03	WARN	C16B	WRITE	C275	WRITE1	C282

BIT Bucket

601 Whitmore Rd., #207
Detroit, MI, 48203
May 21, 1981

Mr. Don Williams, Publisher,
68 MICRO JOURNAL
3018 Hamill Rd.,
Hixson, TN

Dear Don,

I probably told you about these 6809 memory diagnostic programs in one of my earlier letters. I've finally gotten around to cleaning them up (and making a few revisions) and here they're suitable for publication and use by other readers.

Not everyone has a disk system (and therefore, TSC's Diagnostic Package) and for many low-end users, the only memory diagnostic available is possibly a memory diagnostic program in their monitor ROM-- SBug's (SwTPc) 'Q' test, for example. My own system started out as a SwTPc 6800 kit, and the documentation accompanying the kit had source listings for four different diagnostic programs for memory boards. After adding the 6809 cpu to my chassis though (and BEFORE I got the TSC Diagnostic Package, with its battery of memory tests), I had only one memory diagnostic--'Q'.

I resolved to translate those four original 6800 programs to 6809 equivalents. This has taken place over several stages, the culmination being the following four programs. They've been (re-)written to take advantage of some uniquely 6809 features. They're position-independent, first of all. If ANY program should be position-independent, a MEMORY DIAGNOSTIC should be! The latest improvement also makes them ROM-able, so those of you with ROM sockets on your cpu board (or elsewhere) can put these little tests in ROM, if so desired.

A point worth mentioning is that though these versions use the U(ser) Stack for temporary storage, it could just as well be the 'system' stack (SP), and in both cases, it may be necessary to INITIALIZE THE STACK before running the programs. If the US started out containing $0000, for example, the programs would try to use the area $FFFx as WRITE-able storage, something I don't think most of us have.

As I've found myself, little programs like these can be constantly revised and improved, and I'm sure many MICRO JOURNAL readers can find ways to improve these offerings-- I think they make a good starting point, for the 6809 SS-50 user with SBUG.

Sincerely,

Keith Alexander
(313)-862-3454
9AM-4PM EST


```
*
        NAM     MUDGPH1
*
        TTL     Memory Unit Diagnostic--Ph.1
        STTL    For 6809 w/SBUG
*
        OPT     PAG
*
* Memory Unit DiaGnostic, PHase 1
*
* July 22, 1980
*
* This edition: May 14, 1981
*
* This program is a 6809 version of the 6800 memory diagnostic program
* called MEMCON-3, supplied with the SWTBUG(tm) (6800 ROM) documentation.
* It's an address complement-type test, meant to be used with other
* diagnostic programs to FULLY test a piece of memory hardware.
*
* The test may be stopped any time after it has begun the diagnostic by
* inputting a carriage-return (CR). It will only indicate the DETECTION
* of an error, it will not report the ADDRESS where the error was found
* or the specific NATURE of the problems encountered. One COULD insert
* a SWI instruction at, say, ERROR, and the contents of the XR should
* contain the failing address, and the BR should have the failing
* pattern. This would of course obviate the need for the succeeding
* BRA EXTST (test for exit request) after ERROR.
*
* As written here, the program will not even stop when it encounters an
* error, but just continue looping, printing X's where it gets failures.
*
* It is ROM-able and written in POSITION-INDEPENDENT CODE. It also
* presumes the use of SBug-E(tm) (vn.1.5) monitor ROM, and uses
* low level routines found therein.
*
* It may be relocated anywhere in memory by simply changing the ORG state-
* ment and re-assembling, or using an ORG of $0000 and an offset loader.
* (5-14-81) This revision uses a designated
* area on the (U) stack for local storage
*
* SWTBUG and SBUG are trademarks of Southwest Tech. Products Corp.

        *       EQUATES

FD2B  GETADR  EQU  #FD2B    undocumented routine (JSR)) gets hex addr. into XR

F80A  OUTCH   EQU  #F80A    Output ASCII char. in AR
F806  INCHE   EQU  #F806    Input char. to AR, echo, mask parity
F808  INCHEK  EQU  #F808    Check for char. from control term.
F80C  PDATA   EQU  #F80C    Prints string pointed at by XR (no LF,CR)
F80E  PCRLF   EQU  #F80E    Print LF,CR
F814  EXIT    EQU  #F814    SBug re-entry
      *
      * Local storage implemented on U Stack.
      * Defined by offsets from US
      *
0005  SIZE    EQU  5        Qty. of bytes necessary
```

```
Memory Unit Diagnostic--Ph.1          5-20-81 TSC 6809 ASMB.   PAGE    1
For 6809 w/SBUG

               0000  LOMEM   EQU    0
               0002  HIMEM   EQU    2
               0004  BSTORE  EQU    4

0000                         ORG    $0000     POSITION-INDEPENDENT CODE

0000 0D 0A 53 74     MSG1    FCC    $D,$A,"Start address ? ",4
0004 61 72 74 20
0008 61 64 64 72
000C 65 73 73 20
0010 3F 20 04
0013 0D 0A 45 6E     MSG2    FCC    $D,$A,"End address ? ",4
0017 64 20 61 64
001B 64 72 65 73
001F 73 20 3F 20
0023 04

0024 20    01        MDGPH1  BRA    START

0026 09              VN      FCB    9         (5-14-81)

0027 33    5B        START   LEAU   -SIZE,U   Set aside local storage area
0029 30    8C D4             LEAX   MSG1,PCR
002C AD    9F F80C           JSR    [PDATA]
0030 BD    FD28              JSR    GETADR
0033 AF    C4                STX    LOMEM,U   Save start address
0035 30    8C D8             LEAX   MSG2,PCR
0038 AD    9F F80C           JSR    [PDATA]
003C BD    FD28              JSR    GETADR
003F AF    42                STX    HIMEM,U   Save end address
0041 AD    9F F80E           JSR    [PCRLF]
0045 E7    44        START2  STB    BSTORE,U  Save initial pattern
0047 AE    C4                LDX    LOMEM,U   Get start addr. of range --> XR
0049 E7    84        LOOP1   STB    0,X       Save pattern at address
004B AC    42                CMPX   HIMEM,U   End of range yet?
004D 27    05                BEQ    CHECK     Yes, go back and check data
004F 30    01                LEAX   1,X       No, bump to next address
0051 5C                      INCB             Increment pattern
0052 20    F5                BRA    LOOP1     Loop
0054 E6    44        CHECK   LDB    BSTORE,U  Fetch initial pattern
0056 AE    C4                LDX    LOMEM,U   Fetch start address
0058 E1    84        LOOP2   CMPB   0,X       Compare.
005A 26    09                BNE    ERROR
005C AC    42                CMPX   HIMEM,U   O.K. End of range yet?
005F 27    0D                BEQ    CYCLE     Yes, go indicate successful pass
0061 30    01                LEAX   1,X       Else, bump pointer
0063 5C                      INCB             Bump pattern, too
0064 20    F3                BRA    LOOP2     Go loop
0066 86    58        ERROR   LDA    #'X       ** SWI here for stop on first error?
0067 AD    9F F80A           JSR    [OUTCH]   ** Presently just prints an "X"
006B 20    07                BRA    EXTST     ** Go test for CR (exit)
006D 86    23        CYCLE   LDA    #'#
006F AD    9F F80A           JSR    [OUTCH]   "#" indicates successful cycle through range
0073 5A                      DECB
0074 AD    9F F808   EXTST   JSR    [INCHEK]  Check for input from terminal
0078 27    CB                BEQ    START2    No input, continue.
007A AD    9F F806           JSR    [INCHE]   Yes, input received
007E 81    0D                CMPA   #$D       Was it a CR?
0080 26    C3                BNE    START2    No, continue
0082 33    45                LEAU   SIZE,U    Else, Restore stack pointer
0084 7E    F814              JMP    EXIT      and quit.

                             END    MDGPH1

0 ERROR(S) DETECTED
```

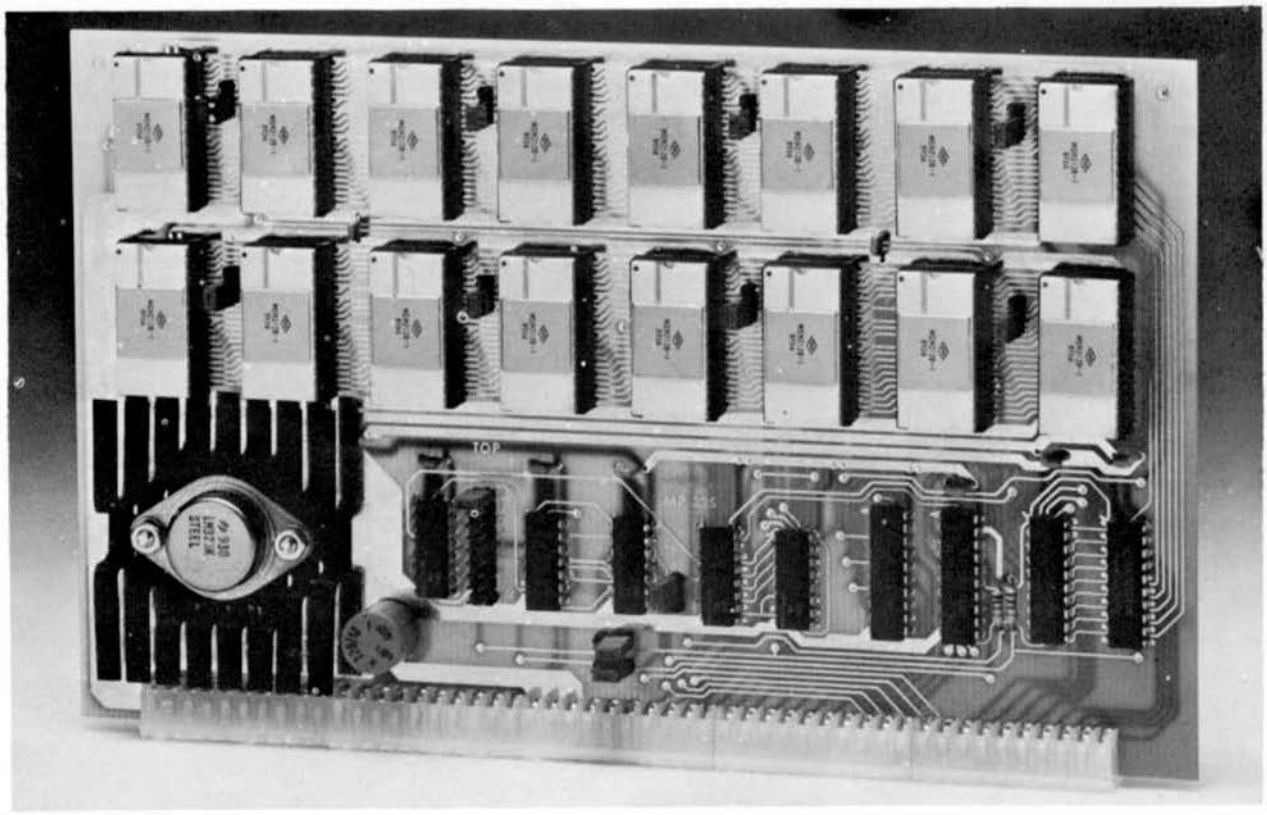
```
*
        NAM     MUDGPH2
*
        TTL     Memory Unit Diagnostic--Ph.2
        STTL    FOR 6809 W/SBUG
*
        OPT     PAG
*
* July 24, 1980
*
*
* This editing: Nov 14, 1981
*
* This program is a 6809 version of the 6800 memory diagnostic program,
* ROBIT-2, supplied with the SWTBUG(tm) (6800 ROM) documentation. It
* is a "walking bit" type diagnostic, moving one bit through each
* bit position of each address under test and reading back what was
* actually written.
*
* Successful passes through the range under test are indicated by plus
* (+) signs, and failures print out the failing address, the test value,
* and the actual contents read back from the failing location.
*
* The program doesn't stop even after a failure; it reports and continues.
* It will stop upon receiving a carriage-return from the user, though.
*
* Meant to be used with OTHER diagnostic programs to FULLY test memory.
*
*
* (5-14-81) This revision uses stack-allocated variable storage.
* Makes program ROM-able.
*
* SBUG and SWTBUG are trademarks of Southwest Tech. Products Corp.
*

*       EQUATES

FD00  OUTS    EQU   $FD00    Output a space ($20)
FD72  OUT2H   EQU   $FD72    Print actual hex contents of Acc. A
FD6A  OUT4H   EQU   $FD6A    Print actual hex contents of XR
FD28  GETADR  EQU   $FD28    Input hex addr from term. to XR
F80A  OUTCH   EQU   $F80A    Output ASCII char. in AR
F806  INCHE   EQU   $F806    Input chr. to AR, echo, mask parity
F808  INCHEK  EQU   $F808    Check for input from terminal
F80C  PDATA   EQU   $F80C    Print string pointed at by XR
F80E  PCRLF   EQU   $F80E    Print, line feed, carriage return
F814  EXIT    EQU   $F814    Or JSR [F800] (indir. SBUG re-entry)
```

```
*
* Local storage expressed as offsets from
* (U)stack. Space reserved at start and restored
* at exit of program
*
0004  SIZE    EQU   4         Qty of bytes needed
0000  LOMEM   EQU   0         Start address of range under test
0002  HIMEM   EQU   2         End address of range under test

0000                         ORG    $0000     POSITION INDEPENDENT CODE, ROM-ABLE

0000 0D 0A 53 74     MSG1    FCC    $D,$A,"Start address ? ",4
0004 61 72 74 20
0008 61 64 64 72
000C 65 73 73 20
0010 3F 20 04
0013 0D 0A 45 6E     MSG2    FCC    $D,$A,"End address ? ",4
0017 64 20 61 64
001B 64 72 65 73
001F 73 20 3F 20
0023 04

0024 20    01        MDGPH2  BRA    GETRNG

0026 09              VN      FCB    9         (5-14-81)

0027 33    5C        GETRNG  LEAU   -SIZE,U   Set aside stack storage area
0029 30    8C D4             LEAX   MSG1,PCR
002C AD    9F F80C           JSR    [PDATA]
0030 BD    FD28              JSR    GETADR    Input low addr. of range under test
0033 AF    C4                STX    LOMEM,U   Save it.
0035 30    8C D8             LEAX   MSG2,PCR
0038 AD    9F F80C           JSR    [PDATA]
003C BD    FD28              JSR    GETADR    Input high addr. of range under test
003F AF    42                STX    HIMEM,U   Save it.
0041 AD    9F F80E           JSR    [PCRLF]
0045 AE    C4        START2  LDX    LOMEM,U
0047 86    01        LODREG  LDA    #1        Store "1" (0000 0001)
0049 A7    84                STA    0,X       at address under test
004B A1    84                CMPA   0,X       Was "1" written?
004D 26    0D                BNE    ERRPNT    No, go report error
004F 48              LOOP1   ASLA             Shift test VALUE left one bit (0000 0010)
0050 68    84                ASL    0,X       Do same to data at test ADDRESS
0052 A1    84                CMPA   0,X       Compare them
0054 26    06                BNE    ERRPNT    If unequal, go report error
0056 81    80                CMPA   #%[illegible] Has test value been fully shifted?
0058 26    F5                BNE    LOOP1     No, go continue ROtating Bits
005A 20    23                BRA    INCR1     Yes, go test next location
005C 1F    12        ERRPNT  TFR    X,Y       Save failing address
005E 1F    89                TFR    A,B       Save failing test VALUE
0060 AD    9F F80E           JSR    [PCRLF]
0064 30    A4                LEAX   0,Y       Failing ADDRESS--> XR (copy in YR)
0066 BD    FD6A              JSR    OUT4H     Print it(XR)--FAILING ADDRESS
0069 BD    FD00              JSR    OUTS      and space
006C 1F    98                TFR    B,A       Test VALUE--> Acc. A (copy in B)
006E BD    FD72              JSR    OUT2H     Print it(A)--Test VALUE (tried)
0071 BD    FD00              JSR    OUTS      and space
0074 A6    A4                LDA    0,Y       Get data actually stored at test ADDRESS
0076 BD    FD72              JSR    OUT2H     Print it, (A)
0079 AD    9F F80E           JSR    [PCRLF]
007D 30    A4                LEAX   0,Y       Point XR at failing ADDRESS
007F AC    42        INCR1   CMPX   HIMEM,U   Compare to end address
0081 27    04                BEQ    FINISH    Branch if done
0083 30    01                LEAX   1,X       Else, bump test ADDRESS by one
0085 20    C0                BRA    LODREG    and continue
0087 86    2B        FINISH  LDA    #'+
0089 AD    9F F80A           JSR    [OUTCH]   "+" indicates successful pass through range
008D AD    9F F808   EXTST   JSR    [INCHEK]  Check for input from terminal
0091 27    B2                BEQ    START2    No, continue test
0093 AD    9F F806           JSR    [INCHE]   Else fetch char. to Acc. A
0097 81    0D                CMPA   #$D       Was it a CR?
0099 26    AA                BNE    START2    No, continue test
009B 33    44                LEAU   SIZE,U    Else, restore stack
009D 7E    F814              JMP    EXIT      then exit program

                             END    MDGPH2

0 ERROR(S) DETECTED
```

```
        NAM     MUDGPH3
*
        TTL     Memory Unit Diagnostic--Ph.3
        STTL    FOR 6809 W/S-BUG
*
        OPT     PAG
*
* July 26, 1980
*
* This editing: Nov 14, 1981
*
* This program is a 6809 version of the 6800 memory diagnostic program
* CDAT-2, supplied with the SWTBUG(tm) (6800 ROM) documentation.
* The original 6800 version was written by John Christensen.
*
* It tests each location of the range under test by first writing all
* 0's to each location, then writing all ones to the first location,
* and then checking each other location to see that they're still 0.
* If so, the 'ones' pattern is replaced with zeroes. Then the next loc-
* ation is written to ones and all other addresses are again checked
* for zero.
*
* There are four types of errors reported, via the SWI (Software Int-
* errupt) instruction, and the address and type of failure are contained
* in the CPU registers when the SWI register-dump occurs. A passing cond-
* ition is reported also.
*
* If a register dump occurs, the IX (X index register) contains the add-
* ress under test when the failure occurred, and the PC will contain
* the address of ERPNT1, ERPNT2, ERPNT3, or ERPNT4, depending on the
* type of error encountered. See comments in source listing below for
* guide to what ERPNTx corresponds to which type of error. For example,
* if a register dump shows 'PC=00B7', it means an error was encountered
* when trying to write the second test pattern (ones) to an address.
* If the error is a dual address error, indicating writing to one addr-
* ess is effecting another address, the IX register will contain the
* address to which the test pattern was written, and the IY (Y index)
* will contain the address that was erroneously effected by the write.
*
* For a successful test, a report of "ATP" indicates All-Tests-Passed.
*
* (5-14-81) This revision uses stack-allocated storage
* for local variables. The space is reserved
* and returned within the program.
```

```
* Program is POSITION-INDEPENDENT and ROM-able.
*
* SBUG and SWTBUG are trademarks of Southwest Tech. Products Corp.
*

*       EQUATES

F00C  PDATA   EQU   #F00C     Print string pointed at by XR
FD20  GETADR  EQU   #FD20     Get 16-bit hex addr. to XR
F00E  PCRLF   EQU   #F00E     Print LF/CR



Memory Unit Diagnostic--Ph.3        5-21-81 TSC 6809 ASMB.  PAGE    1
FOR 6809 W/S-BUG

              F814  EXIT    EQU   $F814     SBug re-entry point
                    *
                    * Local storage expressed as offsets from
                    * (U)stack.
                    *
              0004  SIZE    EQU   4         Quantity of bytes reserved
              0000  LOMEM   EQU   0
              0002  HIMEM   EQU   2


0000                        ORG   $0000     POSITION-INDEPENDENT CODE

0000 0D 0A 53 74  MSG1    FCC   $D,$A,"Start address ? ",4
0004 61 72 74 20
0008 61 64 64 72
000C 65 73 73 20
0010 3F 20 04
0013 0D 0A 45 6E  MSG2    FCC   $D,$A,"End address ? ",4
0017 64 20 61 64
001B 64 72 65 73
001F 73 20 3F 20
0023 04
0024 0D 0A 69 6E  MSG3    FCC   $D,$A,"In progress...",4
0028 20 70 72 6F
002C 67 72 65 73
0030 73 2E 2E 2E
0034 04
0035 0D 0A 4D 65  MSG4    FCC   $D,$A,"Memory test PH 3  ATP",4
0039 6D 6F 72 79
003D 20 74 65 73
0041 74 20 50 48
0045 20 33 20 20
0049 41 54 50 04

0040 20  01       MDGPH3  BRA   GETRNG

004F 09           UN      FCB   9         (5-14-81)

0050 33  5C       GETRNG  LEAU  -SIZE,U   Set up stack storage
0052 30  8C AB            LEAX  MSG1,PCR
0055 AD  9F F00C          JSR   [PDATA]
0059 BD  FD20             JSR   GETADR    Get and save
005C 9F  C4               STX   LOMEM,U   start address
005E 30  8C B2            LEAX  MSG2,PCR
0061 AD  9F F00C          JSR   [PDATA]
0065 BD  FD20             JSR   GETADR    Get end address...
0068 30  01               LEAX  1,X       step by one...
006A 9F  42               STX   HIMEM,U   then save.
006C 30  8C B5            LEAX  MSG3,PCR
006F AD  9F F00C          JSR   [PDATA]   "In progress..."

0073 9E  C4       START   LDX   LOMEM,U
0075 4F                   CLRA              0 --> Acc.A

0076 A7  84       LOOP1   STA   0,X       0 --> address under test
0078 A1  84               CMPA  0,X       0 written?
007A 26  38               BNE   ERPNT1    No, so report error 1
007C AC  42               CMPX  HIMEM,U   End of range set?
007E 27  04               BEQ   LOOPAT    Yes, go to next test
0080 30  01               LEAX  1,X
0082 20  F2               BRA   LOOP1     No, continue
0084 AE  C4       LOOPAT  LDX   LOMEM,U
0086 C6  FF               LDB   #$FF      Second test pattern-- all ones
0088 E7  84       LOOP4   STB   0,X
008A E1  84               CMPB  0,X       1's written?
008C 26  29               BNE   ERPNT2    No, go report error 2
008E 1F  12       CHG_DW  TFR   X,Y       Save address-under-test
0090 AC  C4       LOOP2   CMPX  LOMEM,U   Is this first address?
0092 27  08               BEQ   CHECK1
0094 30  1F               LEAX  -1,X
0096 A1  84               CMPA  0,X
0098 26  28               BNE   ERPNT3
009A 20  F4               BRA   LOOP2
009C 1F  21       CHECK1  TFR   Y,X
009E AC  42               CMPX  HIMEM,U
00A0 27  1E               BEQ   FINISH
00A2 30  01       LOOP3   LEAX  1,X
00A4 A1  84               CMPA  0,X
00A6 26  15               BNE   ERPNT4
00A8 AC  42               CMPX  HIMEM,U
00AA 26  F6               BNE   LOOP3
00AC 1F  21       RESTRE  TFR   Y,X
00AE A7  84               STA   0,X
00B0 30  01               LEAX  1,X
00B2 20  D4               BRA   LOOP4
                    *
                    *
00B4 33  44       ERPNT1  LEAU  SIZE,U    First, restore U stack.
00B6 3F                   SWI             Error on initial pattern (writing 0's)
                    *
                    *
00B7 33  44       ERPNT2  LEAU  SIZE,U
00B9 3F                   SWI             Error on second pattern (writing 1's)
                    *
                    *
00BA 33  44       ERPNT3  LEAU  SIZE,U
00BC 3F                   SWI             Dual address error low
                    *
                    *
00BD 33  44       ERPNT4  LEAU  SIZE,U
00BF 3F                   SWI             Dual addres error high

00C0 30  8D FF71  FINISH  LEAX  MSG4,PCR  "Memory test PH 3 ATP"
00C4 AD  9F F00C          JSR   [PDATA]
00C8 33  44               LEAU  SIZE,U    First, restore stack
00CA 7E  F814             JMP   EXIT      then exit
```

```
*
*       MAP     ALGPH
*
        TTL     Memory Unit Diagnostic--Ph.4
        STTL    FOR 6809 W/SBUG
*
        OPT     PRC
*
* July 28, 1980
*
* This edition: Nov 14, 1981
*
* This program is a 6809 version of the 6800 memory diagnostic program
* called SUMTEST2, supplied with the SWTBUG(tm) (6800 ROM) documentation.
* It is a DATA=ADDRESS-type test. This is a fairly common type of memory
* test that rather that putting some common test byte at each location
* tested, puts a pattern determined by the ADDRESS under test. In this
* case, the data is the sum of a pass counter, the MS byte of the
* address, and the LS byte of the address.
*
* The original 6800 version was written by Chris Courtois of MOTOROLA, Inc.
*
* Errors are reported with the pass number, failing bits, and the
* address.
* The user may stop the program at any time with a carriage-return.
*
* (5-14-81)) This revision uses the area
* pointed to by the US for local storage.
* The space is reserved and returned by the program
* Program is POSITION-INDEPENDENT and ROM-able.
*
* SBUG and SWTBUG are trademarks of Southwest Tech. Products, Inc.
*

*       EQUATES

F006  INCHE   EQU   $F006     Input chr. to XR, echo, mask parity
F009  INCHEK  EQU   $F009     Check for input from terminal.
F00A  OUTCH   EQU   $F00A     Output ASCII char. in AR
F00E  PCRLF   EQU   $F00E     $F8+x indicates vector (reg. indir.addr.)
F00C  PDATA   EQU   $F00C     Print string pointed at by XR
FD20  INCADR  EQU   $FD20     Input two 16-bit addrs. to [Y, ]X, resp.
FD2B  GETADR  EQU   $FD2B     Input addr to XR (JSR or LBSR)
FD6A  OUT4H   EQU   $FD6A     Print contents of XR as 4 hex dig.
FD72  OUT2H   EQU   $FD72     Print contents of AR as 2 hex dig.
FDDD  OUTS    EQU   $FDDD     Output a space (&20)
F814  EXIT    EQU   $F814     Or $F814 (dir.) or $F8A0 (vect.) in SBUG
*
* Local storage, expressed as offsets from US.
*
0000  SIZE    EQU   0         Qty. of bytes required
0000  CTR     EQU   0
0001  STORE   EQU   1         XOR of stored data with retrieved data
0002  TEMPX   EQU   2
0004  LOMEM   EQU   4
0006  HIMEM   EQU   6


0000                        ORG   $0000     POSITION-INDEPENDENT CODE

0000 0D 0A 41 64  MSG1    FCC   $D,$A,"Address range to test? ",4
0004 64 72 65 73
0008 73 20 72 61
000C 6E 67 65 20
0010 74 6F 20 74
0014 65 73 74 3F
0018 20 04


001A 20  01       MDGPH4  BRA   BEGIN

001C 09           UN      FCB   9         (5-14-81)

001D 33  50       BEGIN   LEAU  -SIZE,U   Set aside local storage area
001F 6F  41               CLR   STORE,U
0021 6F  C4               CLR   CTR,U
0023 30  8C 0A            LEAX  MSG1,PCR
0026 AD  9F F00C          JSR   [PDATA]   "Address range to test ?"
002A BD  FD20             JSR   INCADR    Start-->XR, End-->YR
002D 30  01               LEAX  1,X
002F AF  46               STX   HIMEM,U
0031 10AF 44              STY   LOMEM,U   YR contains start addr from here on
0034 AD  9F F00E          JSR   [PCRLF]

0038 1F  21       START   TFR   Y,X       Start addr-->XR
003A BD  20       LOOP1   BSR   INCRX
003C A7  84               STA   0,X       XR points to R.U.T.
003E 30  01               LEAX  1,X       Base to next address
0040 AC  46               CMPX  HIMEM,U   End of range set?
0042 26  F6               BNE   LOOP1     No, continue
0044 AE  44               LDX   LOMEM,U
0046 8D  14       LOOP2   BSR   INCRX     Form data again
                    *
0048 A8  84               EORA  0,X       bits set if any discrepancy w/original
                    *
004A 26  19               BNE   ERROR     No, so report error
004C 30  01       RETURN  LEAX  1,X       Yes, bump to next loc.
004E AC  46               CMPX  HIMEM,U
0050 26  F4               BNE   LOOP2
0052 86  20               LDA   #' '
0054 AD  9F F00A          JSR   [OUTCH]
0058 6C  C4               INC   CTR,U     One successful pass through range
005A 2B  29               BMI   EXTST     Test for operator interject (Quit?)
005C 9F  42       INCRX   STX   TEMPX,U   Save Address Under Test
005E A6  42               LDA   TEMPX,U   MSB-->Acc.A
0060 AB  43               ADDA  TEMPX+1,U Add LS Byte

0062 AB  C4               ADDA  CTR,U     Add pass no.
0064 39                   RTS
0065 A7  41       ERROR   STA   STORE,U   Save errant bit
0067 AD  9F F00E          JSR   [PCRLF]
006B A6  C4               LDA   CTR,U
006D BD  FD72             JSR   OUT2H     Print pass no.
0070 BD  FDDD             JSR   OUTS      Space
0073 A6  41               LDA   STORE,U
0075 BD  FD72             JSR   OUT2H     Print failing bit
0078 BD  FDDD             JSR   OUTS      Space
007B AE  42               LDX   TEMPX,U
007D BD  FD6A             JSR   OUT4H     Print failing address
0080 BD  FDDD             JSR   OUTS
```



UNIVERSAL STATIC MEMORY

★ 32K bytes-ROM, RAM, EPROM or a combination
★ SS-50 A&C compatible with 16 and 20 bit address decoding
★ Compatible with all SWTPC 6800 and 6809 computers
★ 2.0 MHz - 5.0 Volts only

This is the most versatile memory card you can buy. Our S-32 may be populated with up to 32K of static RAM, EPROM, or ROM, or any 4K block combination of these that you may desire. Any 5-volt 2716 pinout compatible memory may be used in this card. Any 4K block of memory may be jumper block programmed for RAM or ROM use. This feature makes this the ideal memory for those process control applications that require a mixture of ROM and RAM memory. The board is fully compatible with all SWTPC 6800 and 6809 computers.

The power requirement for the board is only 1.75 amps at 5.0 volts with a full 32K of RAM installed.

S-32 Circuit card only . . . . . $ 99.50
S3216 with 16K of RAM . . . . . $295.00 ea.
S3232 with 32K of RAM . . . . . $495.00 ea.

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

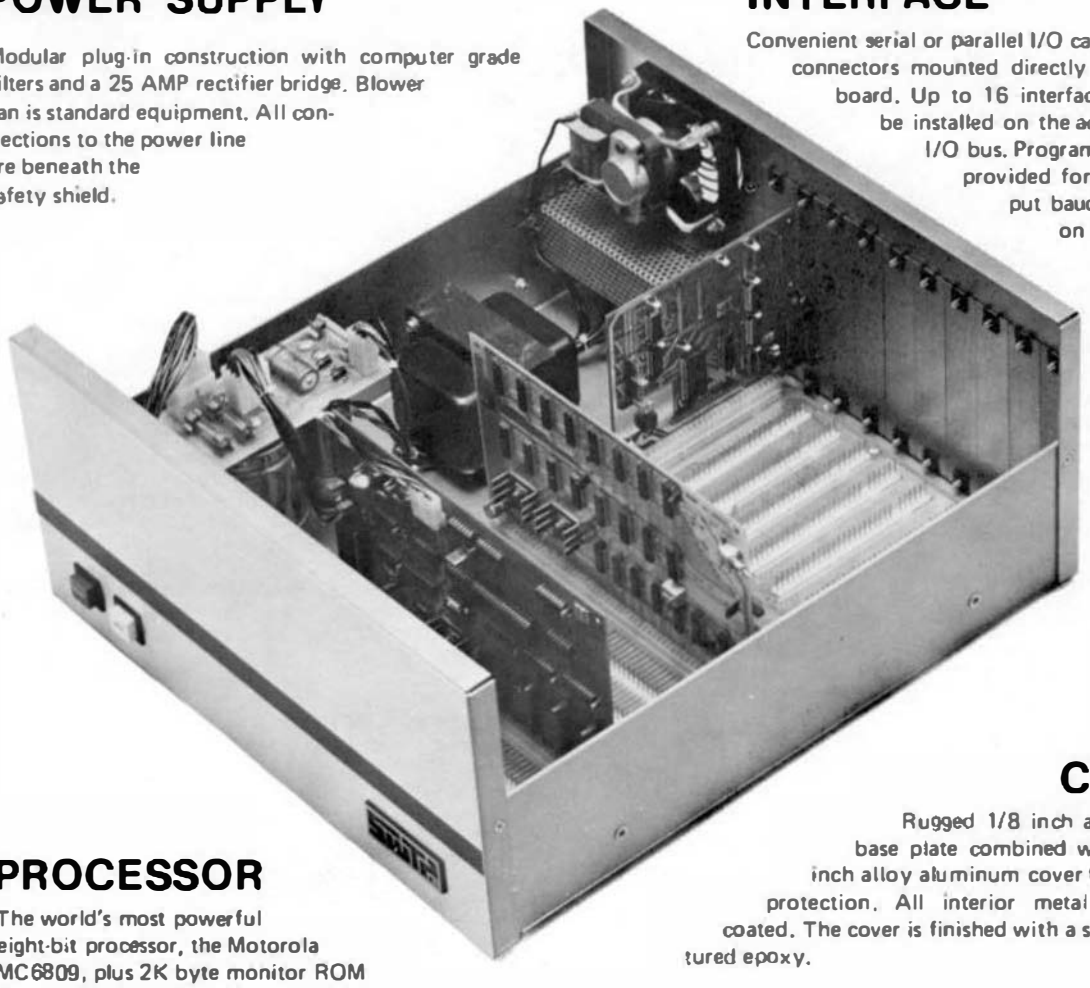
WE HAVE A 6809 FOR YOU

POWER SUPPLY

Modular plug-in construction with computer grade filters and a 25 AMP rectifier bridge. Blower fan is standard equipment. All connections to the power line are beneath the safety shield.

INTERFACE

Convenient serial or parallel I/O cards have DB-25 connectors mounted directly on the circuit board. Up to 16 interface devices may be installed on the address decoded I/O bus. Programming strips are provided for input and output baud rate selection on each port. All outputs are fully buffered.



CABINET

Rugged 1/8 inch alloy aluminum base plate combined with a solid 1/8 inch alloy aluminum cover for unsurpassed protection. All interior metal is conversion coated. The cover is finished with a super tough textured epoxy.

PROCESSOR

The world's most powerful eight-bit processor, the Motorola MC6809, plus 2K byte monitor ROM that is 2716 EPROM compatible and full buffering on all output lines. Built-in multiuser capability, just add I/O cards to operate a multi-terminal system.

MEMORY— You can purchase the computer with either 8K bytes of RAM memory (expandable to 56K), or with the "S" series 64K bytes of RAM memory expandable to 768 K.

PERIPHERALS—The wide range of peripheral hardware that is supported by the 6809 includes: dot matrix printers (both 80 and 132 column), IBM Electronic 50 typewriter, daisy wheel printers, 5-inch floppy disk system, 8-inch floppy disk systems and a 16 megabyte hard disk.

SOFTWARE— The amount of software support available for the 6809 is incredible when you consider that it was first introduced in June, 1979. In addition to the FLEX9 operating system, we have a Text Editor, Mnemonic Assembler, Debug, Sort-Merge, BASIC, Extended BASIC, MultiUser BASIC, FORTRAN, PASCAL and PILOT.

69/K Computer Kit with 8K bytes of memory ..... $ 575.00
69/A Assembled Computer with 8K bytes of memory ..... $ 695.00
09/ Assembled Computer "S" series with 64K bytes of memory $1,595.00

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

```
0083 20  C7          BRA   RETURN
0085 AD  9F F000 EXTST JSR   [INCHEK] Check for input from terminal
0087 27  A0          BEQ   START    No char.I continue test
0089 AD  9F F006     JSR   [INCHE]
008F 81  0D          CMPA  #$0D     Carriage return?
0091 26  A5          BNE   START
0093 33  48          LEAU  SIZE,U   If so, restore stack then...
0095 7E  F814        JMP   EXIT

                     END   PROG44

0 ERROR(S) DETECTED
```

PRODUCT ANNOUNCEMENT
HELP
A Data Retrieval Utility

By Dale L. Puckett
and Frank Hogg

HELP is a data retrieval utility command written by '68 Micro Journal Contributing Editor Dale L. Puckett to save your time. No longer will you need to spend hours digging through system manuals looking for information about the many language commands and statements. It is shipped with two data files compiled by Frank Hogg Laboratory, Inc. One covers every FLEX command and the other covers all TSC BASIC and EXTENDED BASIC command. Since HELP resides entirely in the FLEX utility command area it may be called from other programs.

Here is a good example. Imagine you are writing an article about the FLEX operating system and you forget how to divert the output of a command from the terminal to a printer. Since it's a piece of information you need for your article it would be nice to be able to find it without thumbing through a manual. If you are using an editor like STYLOGRAPH which allows you to execute FLEX commands, you need only go to the command mode and type +HELP P (CR). The information you need will appear on the screen and control will be returned to the editor.

Here's another example. Suppose you are writing a BASIC program and you forget the syntax of the PEEK command. You need only type, +HELP PEEK <BASIC (CR) and you will have that information on the screen in front of you. After HELP reports, control will be returned to BASIC.

Besides retrieving information from the two supplied data files, HELP also contains an interpreter which recognizes commands within the data files.

USING HELP

There are three ways to call HELP from FLEX or another program. Typing HELP (CR) will cause the program to print a screen full of information which tells how to use HELP. It then asks you what you would like HELP with?

Typing HELP MEMEND (CR) will cause HELP to search the default data file, HELPFILE.DIR for an occurence of MEMEND. It then prints a definition of the command and gives an example of the syntax which should be used. Typing HELP PEEK <BASIC (CR) will cause HELP to open the data file, BASIC.DIR and search for PEEK. The desired information will then be printed on the terminal.

If you remember the first letter or two of a command but can't remember the entire command name you need only use HELP's wildcard option. For example, if you know your command starts with PR and can't remember the rest, you can type HELP PR? HELP will respond by giving you information about PR, PRINT and PROT. Similarly HELP P? would print information about every command that starts with a P.

HELP DATA FILES

It is easy to prepare personalized data files that contain information you need to have at your finger tips. In fact, we hope you will share any data files you create with other HELP owners. With the new double-sided, double-density disks now on the market and a little typing from a few HELP users, we could all have a very valuable data base at our finger tips.

HELP COMMANDS

HELP recognizes the following commands within a data file: IF, INPUT, INPUT-YN, PRINT-SEARCH, WILDCARD, PRINT-ON, PRINT-OFF and RESTORE.

IF compares the word in a target buffer with the word following the command ">IF" in the data file. If the word is a match, the old data file is closed and the word following ">IF" is moved into the FCB as a filename. That file is then opened for reading.

EXAMPLE: >IF PASCAL

If you type "HELP PASCAL" or answer a prompt with "PASCAL" and the the line above is in the data file, it will cause the 52d data file to be closed and PASCAL.DIR to be opened for reading.

INPUT issues a prompt which follows the command. For example, ">INPUT What do you want HELP with now?" would echo the prompt, "What do you want HELP with now?" to the terminal and then call a routine to enter your reply into the program's target buffer.

INPUT-YN echoes a prompt and calls FLEX's GETCHR routine. If it gets a "Y" or "y" for yes, it prompts the user for another word. If not, it closes all files and exits to FLEX or the calling program.

PRINT-SEARCH is a routine that tells you what HELP has been looking for when a match is not found. It is handy in the case of typos. WILDCARD looks for a match between your target word and anything which follows it in the same line of the data file. If there is a match, the remainder of the line is printed. WILDCARD then returns to the main loop with the print flag set.

PRINT-ON allows the author of a data file to set HELP's print flag. He can then print any messages he deems necessary. PRINT-OFF does just the opposite and clears the print flag within HELP.

RESTORE is a command provided to allow you to rewind the file. It is usually used at the end of a data file and allows HELP to read through a file as many times as required.

HELP is available from FRANK HOGG LABORATORY, INC., 130 Midtown Plaza, 700 East Water St., Syracuse, NY 13210. Phone: 315-474-7856. It sells for $29.95

THE HARD-SOFT CONNECTION

Francis Wassen
8 Cite Strauss
L-LUXEMBOURG /Europe

SUBJECT:How to use two floppy-disk controllers on a same computer, allowing to work with hard-sectored and soft-sectored diskettes.

1.INTRODUCTION:

Three years ago, when floppy-disk drives and their controllers where still an expensive material, I bought a PERCOM LFD400 system with a single drive; for

$595 that allowed me to use a DOS with all its conveniences. Some months later I upgraded by adding a second drive, which greatly eased file managing and file copying. At the same time, the school where I work as a physics teacher acquired a full fledged SWTPC system, including the MF-68 dual drive system with the appropiated disk controller. Now everything would have been for the best, but there remained a formidable snag: the MF-68 system uses the Western Digital 1771 chip to run soft-sectored diskettes, and PERCOM's LFD400 uses a hard-sectoring method: farewell, compatibility !
Fortunately, PERCOM issued in October 1979 a progress package, called TRANSFLEX and sold now under the name of SOFTRAN. This resaveable set of programs allows:

-to copy a FLEX soft-sectored diskette on a hard-sectored one

-to run FLEX software, using a Flex version called PERCOM FLEX, which is the original FLEX tailored to fit the LFD400 controller.

Alas, the way to compatibility was still a one-way road: it was now possible to run FLEX on the PERCOM system, but copying a file from the hard-sectored diskette to a soft-sectored one was still impossible. Upon inquiry PERCOM's president Harold Mauch answered me to replace the SWTPC MF-68 system by a PERCOM LFD400; that was hardly a convincing way out of my troubles!

Finally the solution to the problem was an easy one, but involved some supplementary cost: I bought a MF-68 controller from SWPTC and now two controllers in my computer (a SWTPC 6800 model). This enables me to run the whole set of soft-sectored FLEX software, as well as the still useful PERCOM software.

2.The hardware.

The necessary hardware to connect the drives to two different controllers consists essentially of a switch-board and the connecting cables; no elaborate electronics are involved! (see fig.1 for the overal scheme)
Most of the lines used in the 34-lines flat ribbon cables are open collector lines, and could be simply connected together (wired-OR); I choose, after some trials,to use a 12-vole toggle switch, which permits to disconnect completely the controller not in action. Figure 2 shows how the appropriate signals are located on the PERCOM board-edge connector and on the MF-68 Amphenol-Tuchel (AT) connector.
My original PERCOM LFD400 board permits to select only three different drives, whereas the SWTPC controller has provision for using four drives; having only two drives for the moment, the 3-drive capability was enough for me, hence the 12-uple switch. If you want a fourtieme decoding on the SWTPC board, it will be easy to add a supplementary appropriate switch, for instance.

3. The preliminary work.

Working with the two controllers is very easy: simply toggle the switch to flip from one controller (and one diskette-sort) to the other one.
Basically the transfer process works in the following manner:

-the source-file to be transferred is edited on the hard-sectored disk

-the edited file is down-loaded on a soft-sectored katte, using the WRITE instruction of the TSC EDITOR.

When you boot FLEX there is one complication: FLEX looks for the limit of available RAM, overwriting every memory-location with hex B9. To prevent the destruction of the RAM contents, one has first to make a special-purpose hard-sectored diskette which loads the PERCOMFLEX as well as the original FLEX without doing the normal booting: lets call that diskette the NOBOOTDISK. Here is the way to make it:

1.-Switch to the PERCOM controller
  -put a new, initialized hard-sectored diskette in the second drive
  -load PERCOMFLEX from the system diskette in the usual manner
  -save memory contents from $A100 to $BFFF using the instruction >2/SAVE PFLEX A100 BFFF ADD3

2. Switch to the SWTPC controller.

   Boot the original FLEX (from it's soft-sectored diskette, inserted in the first drive)

3. Switch back to the PERCOM controller and save the original FLEX under PERCOM's DOS to the NOBOOTDISK:
   2/SAVE FLEX A100 BFFF ADD3
   Now this NOBOOTDISK contains 2 FLEX versions: both may be loaded without destroying RAM contents.

That terminates the preliminary work; the NOBOOTDISK will be used each time a transfer from a hard-sectored diskette to a soft-sectored one has to be made.

4.The transfer from Hard-sectored to Soft-sectored.

If the source file to be transferred is a file written under PERCOM's MDOSXPLUS, you first have to make a FLEX-compatible copy using PERCOM's MDOSFLEX program (included in the SOFTRAN package):
+++MDOSFLEX.DSSS.1.HELLO.TXT , where DSSS is the PERCOM format for the drive and start-sector of the source file. If the source file was written under PERCOMFLEX, this translation process can be omitted.

To make the transfer, follow these instructions:

1. Switch to the PERCOM controller.
2. Load the original PERCOMFLEX (if not yet done) in the usual manner.
3. Insert the hard-sectored diskette that contains the source-file to be transferred in the second drive.
   To make things a bit more clear, let's call "HELLO" that file.
4. Load the file with the TSC EDITOR:
   +++EDIT.HELLO
5. Go back to the monitor (for instance by pushing the RESET button). Remove the hard-sectored disk that contained "HELLO" and replace it by the NODOOTDISK.
6. Load from the NOBOOTDISK the FLEX, by typing:
   >2/L FLEX (remember:you are still using the PERCOM controller!)
7. Jump to the warm-start address of the EDITOR by typing >J 0203
8. Switch to the SWTPC controller.
9. Remove the two hard-sectored diskettes; put an original FLEX system diskette (containing also the EDITOR) in the first drive, and an initialised soft-sectored diskette into the working drive.
10. Type the commands:
   /T:WRITE! (You are in the EDITOR: T saves the current line to the top, WRITE! copies the whole file to the disk)
   The EDITOR will ask: WHITE TO TAPE OR DISK (T/D)?
   You answer with D
   Now the EDITOR asks: FILENAME ?
   Give the proper name, for instance HELLO1 (the name should be different from the original file-name!) The EDITOR copies now the file to the soft-sectored diskette.
11. TRANSFERRING IS DONE ! (Have a drink!)

5. Comments.

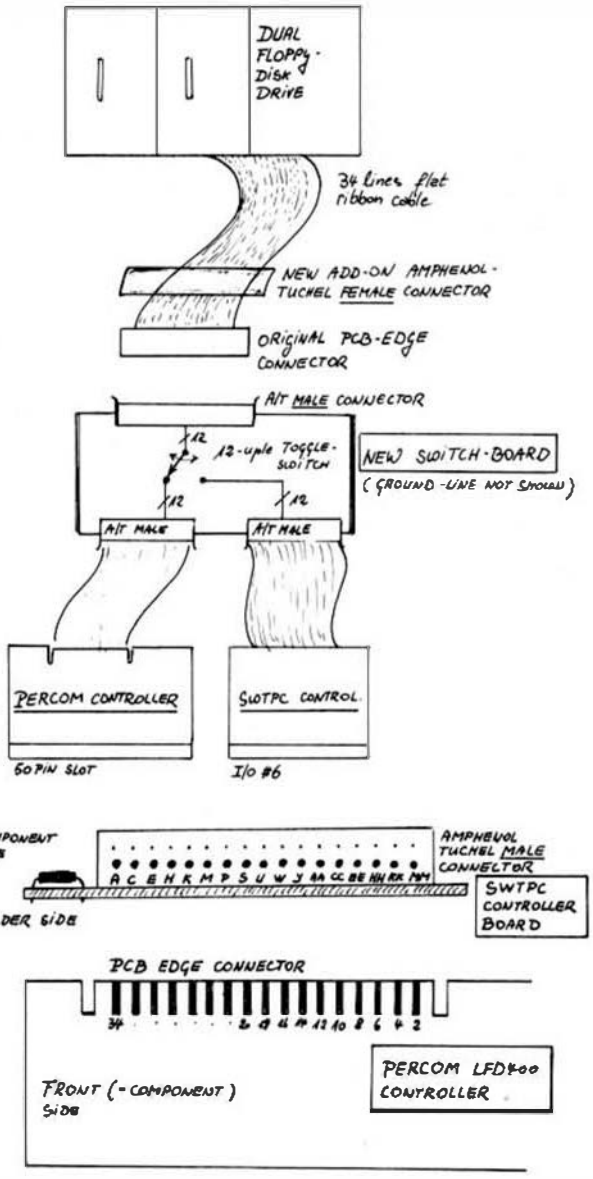
This may seem a somehow messy process, and indeed it is not a very short one. Nevertheless, after becoming acquainted to the operations, a transfer can be done in one or two minutes.
I could not figure out a method to transfer the complete hard-sectored diskette to the soft-sectored one; it will certainly be feasible, and I am grateful for every suggestions.
There is yet another way to do the transfer, and I use this method sometimes for short source-files: make a copy of the file on a punched-paper tape,switch to the SWTPC controller, and input the tape with the EDITOR as if it was typed from the keyboard. My old Olivetti T318 Terminal works on a 110 baud-rate: if the EDITOR's option NUMBER OFF is used, the speed is quite right! However, this is a very time and paper-tape consuming affair, and impratical for large source files.

I use this combination of two floppy controllers for over half a year; they've see not a single problem until now. As the software available in the soft-sectored format is becoming more abundant, I gradually shift away from the hard-sectored format to the other one. Nevertheless the dual-controller-method makes this conversion more painless, and allows me to keep alive some very valuable software from PERCOM.

CORRESPONDANCE TABLE FOR THE FLAT RIBBON LINES

PERCOM		SWTPC	
INDEX	8	IP	EE
DS1	10	DRIVE0	CC
DS2	12	DRIVE1	AA
DS3	14	DRIVE2	Y
NOT AVAILABLE		DRIVE3	KK
MOTOR	16	MOTOR ON	M
DIR	18	DIRC	U
STEP	20	STEP	S
WRITE DATA	22	W DATA	P
WRITE GATE	24	W GATE	N
TRACK0	26	TRK0	K
PROTECT	28	W PROTECT	H
READ DATA	30	R DATA	E

GROUND: ALL ODD LINES GROUND: UPPER PIN ROW



THIS PROGRAM STORES DATA IN MEMORY ON A FIFO BASIS. IT IS USEFUL FOR SUCH THINGS AS PRINT QUEUES AND CRT TERMINAL CONTROLLERS WHERE DISCONTINUITIES IN DATA FLOW CAN OCCUR, SUCH AS INTERRUPTS. BUFFER SIZE IS ARBITRARY AS LONG AS MEMORY IS CONTINUOUS. CAPACITY OF THE BUFFER IS ONE LESS THAN MEMORY BECAUSE OF COINCIDENCE TEST NECESSARY TO DETERMINE IF BUFFER IS FULL OR EMPTY. THE C REGISTER IS USED AS A FLAG IN THIS TEST.

THE OPERATION OF A FIFO BUFFER IS SIMPLIFIED IF ONE VISUALIZES IT AS WRAPPING AROUND ON ITSELF OR AS A COMPLETE CONTINUOUS CIRCLE, WITH BUFFER END ADJACENT TO BUFFER START. A SIMPLE TEST OF BUFFER END TO RESET A POINTER TO BUFFER START JUMPS THE "GAP". YOU WILL NEED FOUR ADDITIONAL BYTES OF MEMORY TO MAINTAIN POINTERS TO THE ADDRESSES OF 'LAST CHARACTER IN' AND 'LAST CHARACTER OUT'.

```
     * EQUATES:
     QIN   EQU  $....  ADDRESS- AST CHAR INTO QUEUE.
     QOUT  EQU  $....  ADDRESS-LAST CHAR OUT OF QUEUE.
     BEGQ  EQU  $....  ADDRESS-START OF BUFFER SPACE.
     ENDQ  EQU  $....  ADDRESS-END OF BUFFER SPACE.
     *
     * INITIALIZE QUEUE ON POWER-ON, RESET, ETC.:
     LDX  #BEGQ
     STX  QIN
     STX  QOUT
     RTS               (OR CONTINUE)
     *
     * PUTQ - PUT CHAR INTO QUEUE. RETURN C=1 FOR
     *        FULL QUEUE, C=0 OTHERWISE.
PUTQ   LDX    QIN       POINT TO LAST INPUT CHAR.
       INX              GO TO NEXT OPEN LOCATION.
       CPX    #ENDQ+1
       BNE    PUTQ1
       LDX    #BEGQ     RESET POINTER ACROSS GAP.
PUTQ1  CPX    QOUT      FULL IF QIN=QOUT,
       BEQ    QFULL
       STA A  0,X       PUT CHAR INTO QUEUE, THEN...
       STX    QIN       UPDATE NEXT INPUT POINTER.
       CLC              SHOW SUCCESSFUL PUT.
       RTS
QFULL  SEC              SET FULL FLAG.
       RTS
     *
     * GETQ - GET A CHARACTER FROM QUEUE. RETURN C=1
     *        FOR EMPTY QUEUE, C=0 OTHERWISE.
GETQ   LDX    QOUT      POINT TO LAST OUTPUT CHAR.
       CPX    QIN       EMPTY IF QOUT=QIN.
       BEQ    QEMPTY
       INX              GO TO NEXT OUTP T LOCATION.
       CPX    #ENDQ+1
       BNE    GETQ1
       LDX    #BEGQ     RESET POINTER ACROSS GAP.
GETQ1  LDA A  0,X       GET CHAR FROM QUEUE, THEN...
       STX    QOUT      UPDATE LAST OUTPUT POINTER.
       CLC              SHOW SUCCESSFUL GET.
       RTS
QEMPTY SEC              SET EMPTY FLAG.
       RTS
```

JIM CARAWAY
11 INWOOD CIRCLE
AUSTIN, TX 78746

DEAR DON,

THERE ARE LOTS OF TRICKS OF THE TRADE THAT GOOD PROGRAMMERS USE THAT ARE UNKNOWN TO MANY OF US NEOPHYTES. LEARNING ABOUT AND USING THESE THINGS IS PART OF THE FASCINATION OF MICROCOMPUTERS. '68MJ' DOES AN OVERALL GOOD JOB IN CATERING TO THOSE OF US THAT ARE NOT PROFESSIONALS. LET'S KEEP IT THAT WAY, AND NOT RUIN AN OTHERWISE GOOD MAGAZINE LIKE THE POWERS AT 'BYTE' DID.

RECENTLY, WHILE REVISING A PROGRAM FOR A CRT TERMINAL CONTROLLER, I CAME ACROSS A ROUTINE TO IMPLEMENT A FIRST-IN FIRST-OUT (FIFO) BUFFER QUEUE SCHEME. I DON'T CLAIM ANY ORGINALITY OR THAT IT IS THE BEST (OR ONLY) WAY TO DO IT. BUT I WOULD LIKE TO SHARE IT WITH YOUR READERS, AND TO SAY THAT I WOULD LIKE TO SEE SIMILAR CONTRIBUTIONS. PERHAPS YOU WOULD LIKE TO CONSIDER THE ADDITION OF A NEW COLUMN FOR "PROGRAMMING TRICKS" TO '68MJ'.

KEEP UP THE GOOD WORK.

FIRST-IN FIRST-OUT (FIFO) BUFFER

Acton CATV, Inc.
Box 407
Acton, Massachusetts 01720
Telephone: (617) 263-7711

May 26, 1981

Mr. Don Williams
'68' Micro Journal
3018 Hamill Road
Hixson, TN 37343

Dear Don:

As you know, one man is solely responsible for the very existance of your magazine and many of our businesses. That man, of course, is Dan Myer, whose innovative skills have created the SS50 bus and continue with new products designed to keep us at the head of the price performance curve.

As you know at the recent dealers meeting, some proposals were somewhat controversal and many new innovative ideas were presented. I am sure he could have followed a much more conservative path emulating other

manufacturers programming. Once again Dan has chosen to lead the pack; and while I doubt that all of his proposals will be enacted in their original form, I am sure he has stimulated our thinking.

Hopefully, his entrepreneurial spirit will rub off on us and make a SWPT dealership the most profitable in the country.

The demonstrations of systems was just a side of amazing, especially the point of sales systems by Omne.

Dan deserves a lot of credit and recognition for his many contributions to our industry and hopefully we will find a way to publicly recognize him.

Sincerely yours,

Ronald A. Mahon
Vice President and General Manager

RAM/pes

Mid South Pharmaceuticals
P.O. Box 584
Hixson, Tennessee 37343
answering 24 hrs
615.842.6597

'68' Micro Journal
3018 Hamill Rd.
PO Box 849
Hixson, TN 37343

Dear Don

Our company has purchased a complete package from Universal Data Research Inc. This package includes the following;

DBM 2 --- Data Base Management Programs
Accts. Receivable/Order Input
Accts. Payable/Purchase Order
General Ledger
Payroll

I placed my order with Joel Heckman at UDRI slightly before the package was finished, but having now received and installed it, I am satisfied that my selection was correct.

I am familiar with some package programs that are very effective in what they are supposed to do but are extremely hard to get anything else out of. With this package from UDRI, in my opinion, the only limitations are imagination. The package provides all the important functions necessary and interacts smoothly and completely as a package. All the files have been carefully designed and are very complete. In addition to performing as a superb package, the DBM 2 allows the untrained user to produce highly specialized functions and reports that are not normally available with a "PACKAGE DEAL". Seeing as how our needs are "special" I highly recommend UDRI's approach to business applications.

Mr. Heckman has been very helpful to us and is easy to reach by phone if there are any problems. His in-house staff are a big boost to the user who has purchased programs before only to find a problem getting it running on their system.

Being a subscriber to your magazine and considering our company as a typical 6809 user, I would recommend that any user considering a solution to their business needs should call Joel and discuss this with him.

In closing I would also like to add a word of praise to your method of business. You are the only magazine I know of that demands so much of their advertisers. I have benefited much from your ability to support these advertisers that meet your demands.

Sincerely,

David Lewis
President

COMPUTERWARE™
Microcomputer Sales and Software

N E W S   R E L E A S E

Computerware® introduces its Color Invaders on cassette for the Radio Shack Color Computer.

You are at the controls of the Color Computer Space Tank, firing at stellar ships and invading critters. Invading ships burst in air with explosive noise. Alien critters march across the screen dropping bombs and screaming as life is zapped from their fried little bodies. Fun?? Yeah!!

With brilliant color, dynamic sound, and fast action, Color Invaders offers a continuous source of excitement to all players. Each of the 8 levels of play present additional complications keeping the beginner going and the experts challenged.

Color Invaders is available in two versions:
Invaders-PP requires 16K and the Power Pack $19.95
Invaders-16 requires 16K (not the Power Pack) $24.95

Both are available directly from Computerware® at Box 668, 1472 Encinitas Blvd., Encinitas, Ca. 92024, (714)-436-3512.

Box 668 • 1512 Encinitas Blvd. • Encinitas, California 92024
Phone: Store (714) 436-0282 • Office (714) 436-3512

A. WELLER
3217 PAGOSA COURT
EL PASO, TEXAS 79904

'68' MICRO JOURNAL
3018 HAMILL RD.
P.O. BOX 849
HIXSON, TENNESSEE 37343

Dear Don,

This is to express appreciation for sharing the "Fix Zero Sectors" item in September, 1980 '68' and to provide and addendum. Since the article was about the SA400 and I have Wangco 82's, I just read it and turned the page.

More recently, I had to run a lengthy program that paused between outputs long enough to let the drives "time out", and couldn't complete a pass without a disk error. Uh-oh. Remembered the article, applied the patch, and back in business. I experimented later and found that a shorter loop (CLRX, DEX, BNE) was enough for my drives and is almost imperceptible.

Please pass on to '68'ers that they should apply that patch no matter what brand of drive, or at least run a test to prove that it's not needed. I'm sure it is the cure for many of our disk failures we've been attributing to "dust particles".

Best regards,

Art Weller

May 30, 1981

STAR-KITS

P O BOX 209
MT KISCO NEW YORK 10549

FOR IMMEDIATE RELEASE

Our HUMBUG Monitor is now available for the Percom SBC/9 Single Board Computer, thereby making this CPU board compatible with FLEX as well as other popular software.

HUMBUG contains all the standard functions and I/O routines of other monitors. But it also includes other functions to make life easier:

- multiple breakpoints
- single-stepping
- memory dump, search, fill and change
- disk boot, tape punch and load
- memory test and memory move
- register examine
- program halt from keyboard

HUMBUG also provides I/O port control, an ABORT function to stop wayward programs and print a register dump, and optional output via a video board.

With this latest version, HUMBUG is now available for 6800, 6802, and 6809 CPU boards made by Gimix, Percom, SWTP, and Star-Kits, and for video boards made by Percom and Thomas. New versions are being developed for other hardware combinations as well. For information send for catalog or call Star-Kits at (914) 241-0287. Turn on your 300-baud modem if you call in the late evening and LIST HUMBUG.DAT.

CER-COMP
5566 Ricochet Ave.
Las Vegas, Nevada 89110
Phn 702-452-0632

June 1,1981

NEW PRODUCTS RELEASE
--- -------- -------

TRS-80 COLOR COMPUTER SOFTWARE

We are now making available for the Color Computer two new programs, both are written in 6809 machine language and are available on compatible TRS-80C machine language cassette tape.

The first of these programs is a Text Editor program which occupies approximately 4K of memory space, including the screen and tape buffer. This leaves almost 12K of user workspace in a 16K system. The Editor can load and save tapes in a format compatible with Basic so that you can easily write and edit Basic programs. The Editor also has a very powerfull set of screen editing commands to allow insertion, deletion, changing and adding to text lines. Automatic line editing allows the user to skip forward and backward in the text buffer for checking and editing. All screen editing immediately updates the screen so you will know exactly what you are doing at all times. The Editor also has commands to move or copy portions of the text buffer from one place to another. Then there is the replace command that allows you to change a single character or string of characters from one thing to another or delete them entirely. Line numbers can also be inserted or deleted for file compression when saving to tape. The Editor also contains two resequence commands one for Basic programs and one for general line resequence. Some of the other commands include Search, Size, Tload, Tappend, Tsave, Auto line, Delete, Printer and Linedit.

The other program is a Co-resident Editor/Assembler that will allow the user to create, edit and assemble machine language programs for the color computer. The editor portion of the program is similar to the text editor described above with all the powerful editing commands except the basic program resequencer. The assembler will output machine object code to either cassette tape in a 'CLOADM' readable format or directly to memory for direct execution. The assembly listing can optionally be output to the printer connected to the RS-232/printer port on the color computer. All error are displayed with a full text message for easy identification. The assembler supports the full compliment of the 6809 instruction set and also will cross assemble 6800 source code to produce 6809 compatible object code.

Prices:	Editor	cassette tape w/manual	$19.95
	CO-RES9	" " "	$39.95
Both	Editor & CO-RES9	tape w/manuals	$49.95

TRS-80 is a trademark of Tandy Corp.

June 13, 1981.
946 Evans Rd.
Nashville, TN 37206

Mr. Don Williams, Sr.
'68' Micro Journal
3018 Hamill Rd.
Hixson, TN 37343

Dear Sir:

Here is a way to force SWTPC Mini-Flex Basic 3.0 to accept commas in string input.

```
0010 POKE( 4522,0)   :REM TURN OFF COMMA BREAK
0020 INPUT A$        :REM INPUT COMPLETE LINE INTO A$
0030 POKE( 4522,44)  :REM TURN ON COMMA BREAK
0040 PRINT A$        :REM PRINT COMPLETE LINE FROM A$

READY
#RUN
? THIS, IS, A, TEST.
THIS, IS, A, TEST.

READY
#
```

This will work with disk reads as well as input from the terminal. For later versions of SWTPC Mini-Flex Basic 3.0, the address to poke is 4527 (This courtesy of Mr. P. O. Marchais). What Basic does is to scan the input line for a comma and break the input string at that point. The POKE(4522,0) or POKE(4527,0) changes the separator character from a comma to a null. Because the null won't be part of an input string (the input character routine filters out nulls), the test fails and the whole line goes into the input variable. The poke makes SWTPC Basic behave like a Basic with an INPUT LINE command. All the POKE(4522,44) does is to replace the comma break. To find out which poke address to use, look for a CMP A #$2C (81 2C) at $11A9 or $11AE. The operand in the first case is at decimal 4522. In the second, it is at decimal 4527.

This should be handy for anyone who wants to process text with SWTPC Disk Basic 3.0. Perhaps someone can find the same location for DMAF-1 Basic 3.5.

Very truly yours,

W R Hamblen

William R. Hamblen

May 26, 1981

Larry Williams
68 Micro Journal
3018 Hamill Rd. Box 849
Hixon, Tenn. 37343

Dear Mr. Williams,

A gremlin crept in and switched two lines of code in my patch to make Dynasoft Pascal Rel. 1.2 work with Flex.

The result messed up the save portion of my patch. The program will load 8.K. and is usable if you use the T.S.C. editor to create and edit source files.

To correct this error (see June issue for listing), make the following changes:

```
06A9  BD  AD  18        JSR INBUFF

06AC  CE  A8  40        LDX #FC8
```

This will make the save function work as advertised.

Garry O. Caudell

Garry O. Caudell

technical systems consultants, inc.

P.O. Box 2570 • 1208 Kent Avenue • West Lafayette, Indiana 47906 • (317) 463-2502

6809 NATIVE-CODE PASCAL COMPILER
FOR FLEX™ AND UniFLEX™

The demand for a higher level language that produces fast and efficient code has prompted Technical Systems Consultants, Inc. to develop a 6809 Native-code Pascal Compiler. This Pascal compiler produces actual 6809 assembly language mnemonics, unlike many of the other Pascal "compilers" which only produce interpretive "P-Code". Because of this native-code production, programs developed with the Native-code Pascal Compiler may run from 5 to 10 times faster than those compiled using an interpretive compiler.

The specification for the syntax and semantics of Pascal for this compiler are based on the Jensen and Wirth User Manual; the compiler implements nearly all of the features defined in the User Manual. The few exceptions of major features that our Native-code Pascal Compiler does not support includes GOTO statements and labels, procedures and functions used as parameters, the nesting of procedures and functions without the use of FORWARD, and the procedures DISPOSE, PACK and UNPACK.

Both integer and floating point math are supported. The floating point arithmetic is double precision containing up to 16.8 digits of accuracy from 1.0 E-38 to 1.0 E+38. This compiler supports the standard trigonometric, exponential and square root functions along with a random number generator for statistical and simulation programming. Integers range from -32768 to +32767, using 16 bits for each integer. All of the ASCII characters from 0 to 127 may be used and written in Pascal program.

Variable names are unique to 160 characters allowing the users greater creativity in programming. All upper and lower case letters and numbers are allowed in variable names along with the underscore character (_), except as the first character.

PACKED arrays and records are allowed in the syntax; however, a PACKED array or record is no different than an unpacked array or record. Pascal sets may contain up to 128 elements; however, the ordinal value of these elements must be from 0 to 127. Therefore, sets of real numbers are not implemented, but a set of characters is easily accommodated.
The Native-code Pascal Compiler allows the users to read the command line in both FLEX and UniFLEX through a record structure called PARAM. In this way the user may pass parameters from the command line to the user's Pascal program. The passing of file names and options is a common application of this parameter passing feature. Furthermore, both FLEX and UniFLEX versions of the Native-code Pascal Compiler allow the users to redefine the standard Pascal input and output files as external files residing on the disk.

The Native-code Pascal Compiler supports dynamic storage allocation using the standard procedure NEW and procedures MARK and RELEASE for dynamic deallocation of storage. Pointer type variables are fully supported; therefore, true file I/O using file buffer pointers and the procedures GET and PUT is implemented. FLEX Pascal files are all sequential access; however, the UniFLEX version allows the user to randomly seek to various positions within the file.

The Native-code Pascal Compiler allows a Pascal program to call other separately written and compiled Pascal programs or assembly language programs. Furthermore, parameters may be passed to these other programs in the same fashion that parameters are passed from the command line.

Additional procedures exist for the users to interface with the operating system itself. These routines include SYSTEM_DRIVE and WORK_DRIVE for FLEX, BUFFER and UNBUFFER for single character input, KILL for deleting UniFLEX files, and various other routines tied to the operating systems. The UniFLEX version of Pascal includes a separate systems run-time package for systems programming. Included in this run-time package are the calls to the system routines within the UniFLEX operating system. These calls allow the user to "fork" or "exec" tasks, initiate interrupts, and execute many other system calls included in the UniFLEX operating system.

Instructions for trimming the run-time package for Pascal are included. Trimming the run-time package may be helpful if a program does not need many of the functions but requires a great amount of memory for execution. By trimming the run-time package, a program may be able to reclaim the memory space allotted to the useless run-time procedures.

Overall, the 6809 Native-code Pascal Compiler produces very fast and efficient code. The FLEX version of Pascal requires a 56K system in order to function. The following prices include our user's manual, a copy of the Pascal User Manual and Report, by Jensen and Wirth, the compiler and run-time object code programs and about ten example Pascal programs in source form.

FLEX 6809 Native-code Pascal Compiler	$200.00
Manual only - $40.00	
UniFLEX 6809 Native-code Pascal Compiler	$225.00
One year maintenance - $75.00	
Additional manuals - $20.00 each	
Manual only - $40.00	

```
WORDS                                   6-15-81        PAGE 1

10 REM WORD COUNTING PROGRAM -- COUNTS WORDS AND
20 REM  USAGE OF WORDS IN A TEXT FILE
30 REM  D.K. BECK              5/13/81
40 REM OPEN FILE AND GET FILE NAME (DEFAULT .AT)
50 ON ERROR GOTO 720
   DIM L3$(999),M3%(999)
70 INPUT "FILE NAME ";F1$
80 OPEN OLD F1$ AS 5
90 REM INITIALIZE VARIABLES- F4%=WORD COUNT FLAG
100 REM I6%=GENERAL COUNTER  K1%=SEPARATE WORD COUNTER
110 REM P1%=PAGE NUMBER  W1%=WORD COUNT
120 F4%=1
130 I6%=1:K1%=0
140 P1%=1:W1%=0
150 CR$=CHR$(13):BP$=CHR$(32)
160 INPUT LINE #5,L$
170 IF I6%=LEN(L$)+1 THEN L1$=CR$:I6%=1:GOTO 210
180 L1$=MID$(L$,I6%,1)
190 I6%=I6%+1
200 REM WORD COUNTER LOGIC- KICKS OUT MULTIPLE SPACES BETWEEN WORDS
210 IF ((L1$=CR$ OR L1$=BP$) AND F4%=1) THEN W1%=W1%+1:F4%=0:GOTO 380
220 IF ((L1$=CR$ OR L1$=BP$) AND F4%=0) THEN GOTO 370
230 IF (F4%=0 AND ((L1$ <> CR$ OR L1$ <> BP$)) THEN F4%=1
240 REM DELETE PUNCTUATION MARKS
250 IF ASC(L1$) (= 64 GOTO 170
260 IF (ASC(L1$) )= 92 AND ASC(L1$) (= 96) GOTO 170
270 REM BUILD UP THE NEXT WORD
280 L2$=L2$+L1$:GOTO 170
290 REM LINEAR SEARCH OF CONCORDANCE TABLE FOR WORDS ALREADY FOUND
300 FOR J6%=0 TO K1%
310 IF L2$=L3$(J6%) THEN M3%(J6%)=(M3%(J6%))+1:GOTO 360
320 NEXT J6%
330 REM SAVE A NEW WORD
340 L3$(A1%)=L2$
350 M3%(A1%)=1:K1%=K1%+1
360 L2$=""
370 IF L1$=CR$ THEN GOTO 160 ELSE GOTO 170
380 REM BUBBLE SORT RESULTS
390 REM VARIABLES- F3%=FLAG TO SWAP WORD COUNT  K2%=SORT INDEX
400 REM F2%=FLAG SWAP DONE.  REPEAT UNTIL NO SWAPS
410 F3%=0
420 K2%=K1%-1
430 F2%=0
440 FOR I6%=0 TO K2%
450 IF L3$(I6%+1)="" GOTO 480
460 IF L3$(I6%) ) L3$(I6%-1) THEN SWAP L3$(I6%),L3$(I6%+1):F2%=1:F3%=1
470 IF F3%=1 THEN SWAP M3%(I6%),M3%(I6%+1):F3%=0
480 NEXT I6%
490 IF F2%=1 THEN K2%=K2%-1:GOTO 430
500 REM PRINT RESULTS
510 REM CHECK IF FOR PRINTER OR TERMINAL OUTPUT
520 INPUT "HARD COPY ",H$
530 IF LEFT$(H$,1)="Y" GOTO 560
540 IF LEFT$(H$,1)<>"N" GOTO 520
550 GOTO 570
560 OPEN OLD "0",PRINT" AS 0
570 GOSUB 760
580 PRINT #0,"There are a total of "+M1%+"words in file "+F1$
590 PRINT #0,"There are "+K1%+"different words."
600 PRINT #0
610 PRINT #0,"Word","No. times used"
620 PRINT #0
630 K6%=K6%+5
640 FOR L6% = 0 TO K1%-1
650 PRINT #0, L3$(L6%),M3%(L6%)
660 K6%=K6%+1
670 IF K6%=57 THEN GOSUB 760
680 NEXT L6%
690 PRINT #0,CHR$(12);:PRINT #0,CHR$(12);:REM PAGE UP PRINTER
700 STOP
710 REM ERROR CHECKER.  IF EOF, CLOSE FILE AND SORT RESULTS
720 IF ERR()8 THEN PRINT "ERROR "+ERR$+" AT LINE "+ERL$:STOP
730 CLOSE 5
740 GOTO 410
750 REM PAGE HEADER FOR OUTPUT
760 PRINT #0,CHR$(12);:REM FORM FEED
770 PRINT #0:PRINT#0
780 PRINT #0,"Word Count";TAB(65);"Page "+P1$
790 PRINT #0
800 K6%=4:P1%=P1%+1
810 RETURN
820 END
```


Edgar M. Pass, President
Computer Systems Consultants
1454 Lotts Lane
Conyers, Georgia 30207
Telephone 404-483-1717/4570

Computer Systems Consultants announces the availability of TABULA RASA. This product provides an electronic spreadsheet facility for 6809 systems which can run the TSC Extended Basic Interpreter and the TSC Macro Assembler. It will operate with most serial terminals and memory-mapped video displays which provide at least 16 rows and 64 columns on the screen, and requires at least 56K bytes of RAM.

TABULA RASA is based upon the CSC Full-Screen Display package. An interactive system generation procedure is used to help the user customize the display drivers for a specific terminal/computer configuration. In most cases, this procedure is sufficient to complete the customization with no further effort.

Data is entered into formatted screens representing the logical classifications of information comprising a spreadsheet, as follows:

Report Headings,
Initial Values,
Computational Instructions.

Results are placed into a file representing a fourth classification, called Computed Values. Any of the four classifications of information in a spreadsheet may be displayed on the terminal or on a printer. Any of the first three classifications may be updated relatively independently of one another. Utility programs allow any of the first three classifications to be used as a part of another spreadsheet, and allow a Computed Values file to be used as a part of an Initial Values file for another spreadsheet.

The maximum size spreadsheet which may be processed by TABULA RASA depends upon the amount of work disk available. A single-sided, single-density, mini-diskette would easily contain a spreadsheet of 100 rows by 25 columns, with 100 computational instructions.

The retail price of TABULA RASA is $100.00. This includes the source files for all programs, on mini or full-size diskettes, and the instruction manual. Dealer inquiries are welcome and should be directed to Bud Pass.

ABORT SWITCH FOR 6809

In the January issue of Kilobaud Microcomputing Peter A. Stark describes adding a Motorola style ABORT switch to 6800 based computers which use SWTBUG.

The ABORT switch generates a NMI which interrupts the program being executed, stores the processor state vector on the stack, prints the register dump and returns to the command loop of the monitor. It is rather like pressing RESTART but as informative as encountering a breakpoint. This means that when a program goes into a "silent loop", the ABORT button can be pressed and the contents of all the registers displayed, including of course the address of the next instruction to be executed.

As I have long felt the need for this facility, I decided to add it to my 6809 based comouter which uses S-BUG (as listed in the June 1980 issue of '68' Micro Journal). The NMI is free to be used for the ABORT function as the NMI vector (address $FFFC-D) in S-BUG points to the RESTART routine.

S-BUG already has all the routines that are necessary and in fact only five bytes need be added in order to be able to use them. There is space for this additional code in the message expansion area. The NMI vector transfers control to the added code, where the U register is made to point to the processor state vector stored on the stack. The program then branches to the breakpoint routine where it displays the registers before returning to the command loop. Any of the normal monitor commands may then be executed before returning the program with the "G" command.

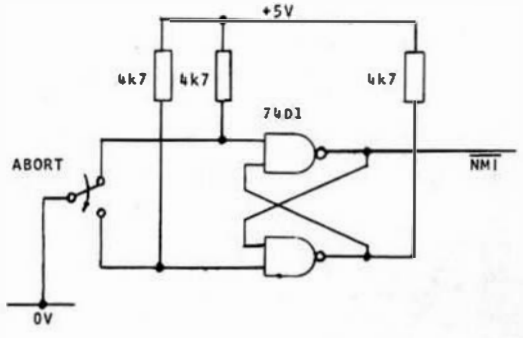
The hardware simply consists of a push button and a bistable to de-bounce it, which is connected to the NMI line.

This simple ABORT function does not work if the contents of the DAT ROM have been changed by the errant program. Forcing the lower bytes of the DAT ROM locations $FFFE and $FFFF to be 1 and 0 respectively could be the solution but I have not yet tried.

D. R. Gaskell, % 21 Beccles Road, SALE Cheshire, M33 3RP
ENGLAND


```
* ABORT SWITCH FOR 6809
* Make the following changes to S-BUG

FEF8   1F 43     ABORT   TFR   S,U
FEFD   16 FB9E           LBRA  REGPR    ($FABE)


FFFC   FE F8             FDB   ABORT    NMI-V
```

CIRCUIT DIAGRAM



HELP

HELP!

Dear Sirs, Reading a back issue (April '80) of the 68'Micro Journal, I came across a SPL/M program to create an index of disk files. It was called WHERESIT.CMD. It was for FLEX. Did anyone do a revised version for Smoke Signal Broadcastings DOS5? If so, could they let me know what it would cost me? PLEASE. I have only a single drive, by the way.

T K Boyd, Belmont School, Feldemore, Holmbury St. Mary, Dorking, Surrey RH5 6LQ England

***

CLASSIFIED

2 SWTP 6800's each with 24K,MPC,MPS,MP-L,JPC Clock 4800 Baud JPC, 2 Cassette Decks, CT-64, Sanyo 9", Fast Basic w/Files Assem, Dis, Much Software $800 each or $1500 for both with PR-40 Printer!
Maine Computer Network 1-207-244-7444

***

5 SWTPC 4K, TI Silent 700 KDB, $35 each. MPA Cpu, Seals 8K, 4-SWTPC 8K, $80 each. 200ns 8K, MPA2 CPU, VAB-2 Terminal, $100 each.
Kent 7-9 PM 1-801-782-8976

***

1 CT-82 CRT terminal (new in box - never used) $845. 1 CT-82 CRT terminal (Used) works fine $745. Decision Data High Speed Dot Matrix 132 Column Tractor Fed Printer (LIKE NEW Quality & Condition) (New cost $2495.00) SAVE $1,000.00 - ONLY $1495.. Leedex Video 100 Monitor (Brand New) $75. SWTPC 5" Double Head Double Density Dual Drive Disk System with DC-3 Controller Board $950 or with DC-4 Controller Board $1050.
LEW 8am-5pm 1-615-877-2241

***

6800 MP-A2 with SWT+DISKbugs $70. Complete 6809 56KB 1MHz with improved PS $975. DMAF-F2 rev C $275. MP-B, MP-82, MP-B2 kit, 6800 cages + PS,MP-LA,MP-S,MP-C,MP-T,MP-32,MP-8, all at 50% of list. Microtime 6800 $75, Percom CIS-30 $65, Hazeltine 1500 $820.
7-9 PM EST 1-603-774-7762

***

SWTPC 4K memory boards (2) Modified for high addressing $45 each. Fully socketed, working, extra 2102's.
Bob Dembinski 21 Spring St,Norfolk,MA 02056 W-(617)671-3045, H-(617)528-3289

***

Infoton GT-100 $500, Microcroma/16K working complete in case with power supply & documentation $400, Practical Automation Printer UP-7 with serial & parallel interfaces in case with power supply & documentation $350.
Bud Pass 1-404-483-1717

***

4 SWTPC 4K memory boards, $40 each or $150 for all four postpaid. PR-40 Printer $200 postpaid.
S Brown, 35 Kettle Pond Road, Amherst Ma 01002, 1-413-253-3183

***

6800 MPA 32K SWATBUG, 9600 Baud Serial, Parallel Boards, SSB 3 Drive 5 1/4" Floppy, Soroc Terminal, System $1500.
Jack PO Box 880, Lafayette In 47902, 1-317-743-4156

***

6809 Color Computer

EDITOR, ASSEMBLER AND MORE!

TIRED OF WAITING FOR SOFTWARE DELIVERY?

WE SHIP FROM STOCK!

NEW!

SOFTWARE DEVELOPMENT SYSTEM

The Micro Works Software Development System (SDS80C) is a complete 6809 editor, assembler and monitor package contained in one Color Computer program pack! Vastly superior to RAM-based assemblers/editors, the SDS80C is non-volatile, meaning that if your application program bombs, it can't destroy your editor/assembler. Plus it leaves almost all of 16K or 32K RAM free for *your* program. Since all three programs, editor, assembler and monitor are co-resident, we eliminate tedious program loading when going back and forth from editing to assembly and debugging!

The powerful screen-oriented Editor features finds, changes, moves, copys and much more. All keys have convenient auto repeat (typamatic), and since no line numbers are required, the full width of the screen may be used to generate well commented code.

The Assembler features *all* of the following: complete 6809 instruction set; complete 6800 set supported for cross-assembly; conditional assembly; local labels; assembly to cassette tape or to memory; listing to screen or printer; and mnemonic error codes instead of numbers.

The versatile ABUG monitor is a compact version of CBUG, tailored for debugging programs generated by the Assembler and Editor. It features examine/change of memory or registers, cassette load and save, breakpoints and more.

SDS80C Price: $89.95

CRACK THOSE ROMS!

SOURCE GENERATOR: This package is a disassembler which runs on the color computer and enables you to generate your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. Disassembler features include cross-referencing of variables and labels; output code which can be reassembled; output to an 80-column printer, small printer or screen; and a data table area specification which defaults to the table boundaries in the interpreter ROM. A 16K system is required for the use of this cassette.

60C Disassembler Price: $49.95

LEARN 6809!

6809 Assembly Language Programming, by Lance Leventhal, contains the most comprehensive reference material available for programming your Color Computer.

Price: $16.95

PARALLEL O!

USE A PARALLEL PRINTER with your Color Computer! Adaptor box plugs into the serial port and allows use of Centronics/Radio Shack compatible printers with parallel interface. Assembled and tested.

Pt80C Price: $69.95

CBUG IS HERE!

MONITOR TAPE: A cassette tape which allows you to:

- Examine or change memory using a formatted hex display
- Save areas of memory to cassette in binary (a "CSAVEM")
- Download/upload data or programs to a host system
- Move the video display page throughout RAM
- Send or receive RS-232 at up to 9600 baud
- Investigate and activate features of your computer, such as hi-res graphics or machine-language music
- Use your color computer as an intelligent peripheral for another computer, a color display or a 6809 program development tool

The monitor has 19 commands in all, and is relocatable and re-entrant.

CBUG Tape Price: $29.95

MONITOR ROM: The same program as above, supplied in 2716 EPROM. This allows you to use the entire RAM space. And you don't need to re-load the monitor each time you use it. The EPROM plugs into the Extended Basic ROM Socket or a modified ROMPACK.

CBUG ROM Price: $39.95

32K RAM!

MEMORY UPGRADE KITS: Consisting of 4116 200ns. integrated circuits, with instructions for installation. 4K-16K Kit Price: $39.95. 16K-32K Kit (requires soldering experience) Price: $39.95

THE MICRO WORKS

MasterCharge/Visa Accepted

California residents add 6% tax.

P.O. BOX 1110, DEL MAR, CA 92014 (714) 942-2400

WESTERN CANADIAN DISTRIBUTOR

Epson Printers
Memorex Disketts
Dealer for:
Gimix
SWTPC
MPI
Alford & Associates
Thomas Instrumentation
Computer Systems Consultants

CUSTOM IN HOUSE SOFTWARE

JIVAL COMPUTER SYSTEMS LTD
12525 A 127 St.
Edmonton, Alberta T5L1A3
Canada (403) 454-0584
Jim Steene

DISK DRIVE WOES?
PRINTER INTERACTION?
MEMORY LOSS?
ERRATIC OPERATION?

Don't Blame The Software!

Pat. #4,259,705

Power Line Spikes, Surges & Hash could be the culprit! Floppies, printers, memory & processor often interact! Our patented ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash.

- ISOLATOR (ISO-1) 3 filter isolate 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket $62.95
- ISOLATOR (ISO-2) 2 filter isolate 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank $82.95
- SUPER ISOLATOR (ISO-3), similar to ISO-1 except double filtering & Suppression $94.95
- ISOLATOR (ISO-4), similar to ISO-1 except unit has 6 individually filtered sockets $108.95
- SUPER ISOLATOR (ISO-11) similar to ISO-2 except double filtering & Suppression $94.95
- CIRCUIT BREAKER, any model (add-CB) ..... Add $ 8.00
- CKT BRKR/SWITCH/PILOT (-CBS) ..... Add $16.00

AT YOUR DEALERS

Master-Card, Visa, American Express
Order Toll Free 1-800-225-4876
(except AK, HI, PR & Canada)

Electronic Specialists, Inc.
171 South Main Street, Natick, Mass. 01760
Technical & Non-800: 1-617-655-1532

VS-1 SPEAKER VOICE SYNTHESIZER...

The VS-1 "SPEAKER" voice synthesizer is an interface board for the SS-50 bus which provides unlimited human speech capability at a modest price. The VS-1 is intended for the industrial and commercial application where specialized vocabularies and ease of programming make "canned-message" synthesizers unsuitable. By using phoneme coding, high quality speech, tailored to meet any application, is possible.

FEATURES –

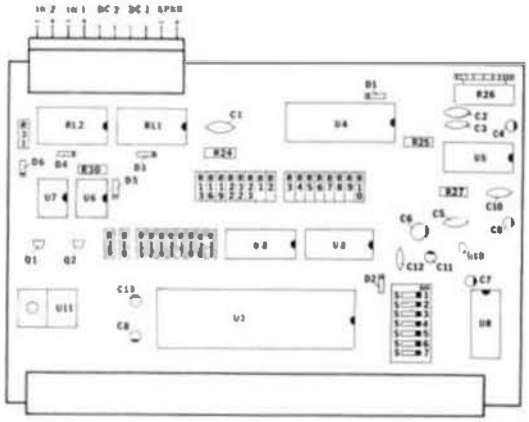
- phoneme coding allows unlimited speech
- low data rate doesn't hog CPU time
- efficient - very little memory needed for messages
- programmable inflection and pitch
- on-board amplifier drives 4- or 8-ohm speaker directly
- assembled, tested, burned in
- meets Class A device radiation limits under FCC Part 15
- dip-switch decode select for 4- or 16-address I/O
- dip-switch programmable NMI and IRQ interrupts

SPECIFICATIONS –

PHONEME CODES	- 61 plus 2 pause, 1 stop
INFLECTION LEVELS	- 4 major, 16 minor - 64 total
VOICE PITCH	- male baritone, variable bass to tenor
DATA RATE	- 4 to 20 BPS, depending upon phoneme
MESSAGE MEMORY	- approx. one byte-per-letter in English
AUDIO OUTPUT	- .2 watt average, 2 watt peak into 8-ohms
RESIDUAL NOISE	- -45 dBm typical
SIZE	- 3 1/2" by 5 1/4" (standard I/O size)
LOCATION	- one 30-pin I/O slot
ADDRESSING	- four addresses in I/O space
POWER REQUIREMENT	- +8 vdc @ 100 ma., +14 vdc @ 150 ma. max.
SPEAKER CONNECTOR	- 10-pin molex, mating half supplied

SOFTWARE –

- VOX-EDITOR macronically edits message (source included)
- example programs in assembly and Basic
- speech files
- shipped on 5- or 8-inch disk for single-user TSC or SSB DOS
- 6800 or 6809 code



VS-1 SPEAKER, manual, program disk $229.95

MASTERCARD, VISA, UPS COD's, checks accepted. All payments MUST be drawn on funds in U.S. banks. Virginia residents add 4% state sales tax. Shipping pre-paid on continental orders. On overseas orders, add $15 for shipping. Open accounts by previous written arrangement only. Shipping from stock to 30 days. Office hours are from 8:30 to 5:30 EST (EDT), Monday through Friday.

ALFORD & ASSOCIATES
P.O. BOX 8743
RICHMOND, VA., 23230
804-320-8722

SMOKE SIGNAL BROADCASTING

31336 VIA COLINAS WESTLAKE VILLAGE, CA 91362 TEL. (213) 889-9340 / TWX 910-494-4965

MICROPRODUCT BULLETIN

THE MOST POWERFUL, FLEXIBLE AND RELIABLE HARD DISK SYSTEMS —

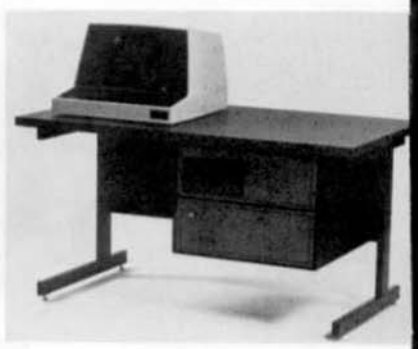
AVAILABLE NOW FROM SMOKE SIGNAL!

- Smoke Signal offers complete Winchester-based systems, all tested for reliability at 2 MHz operation. All CHIEFTAIN™ hard disk computer systems support OS-9 Multi-User operating system and DOS69D.
- All CHIEFTAIN™ Series computer systems — from 5¼-inch to 8-inch to Winchester — are completely compatible.

Smoke Signal's two newest, most exciting additions to the CHIEFTAIN™ Series are the CHIEFTAIN™ 96W10 and CHIEFTAIN™ 96W30. Both of these computer systems provide the 68-00 and 6809 user with state-of-the-art computer performance.

The CHIEFTAIN™ 96W10 — the 10 Megabyte 8-inch Winchester drive computer system — and the CHIEFTAIN™ 96W30 (the 30 Megabyte version) — are available **now** running OS-9 Multi-User and DOS69D.

Under OS-9, the 96W10 and 96W30 systems will run applications provided by Microware (including Stylograph!), Universal Data Research, Washington Computer Services, and any other OS-9-based software. The versatility of the 96W10 and 96W30 extends further with applications from Computerware Software and other DOS69D-compatible applications. Through Smoke Signal's Dealer Information Exchange, hundreds of proven applications are available to Smoke Signal Dealers — all running on the CHIEFTAIN™ Series of computer systems.



High-level languages, including Computerware's Random BASIC, the incredible BASIC 09, and other planned Microware languages, will run on both the CHIEFTAIN™ 96W10 and 96W30 as well as the entire spectrum of CHIEFTAIN™ computers.

The complete hard disk CHIEFTAIN™ systems start at $3696.00. Both CHIEFTAIN™ hard disk systems come Standard with 2 MHz 6809 CPU, 32K of RAM (expandable), dual Serial Ports, DCB-4A Disk Controller Board, DOS69D Operating System, hard disk interface, and one 8-inch double-sided floppy disk drive (1 Megabyte).

Contact Smoke Signal for information on 20 Megabyte Tape Streamer Backup Option.

OKAY . . . I'm interested in your computer

☐ Send CHIEFTAIN™ 96W10 and 96W30 information and product catalog.

☐ Send Dealer Information.

Name_____ Title _____

Company_____

Address _____

City_____ State_____ Zip_____

Phone ( ) _____

SMOOTH™ Software

Modem Program with Disk File Transfer (Instructions and Source Listing) $30.00
Disk with source and object code add 10.00
Specify 6800 or 6809, SSB or FLEX™, 5" or 8"

ALL IN ONE for only $45.00
Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line), input (multiple lines), list, next, overlay (with cursor editing, character deletion and insertion), overstrike (for selected darker text), print, restart, set, top, underline, up, and verify.

Supports Text Processing commands such as block copy, block move, centering, margin justification (wide and narrow), paging, and tabbing.

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally adressed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append one file to the end of another, or insert (merge) one file into another as designated by the line pointer. Print specified lines to your Printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. Delete disk files from the editor.

Printer commands. Control characters can be sent to the printer for format control either directly from the control terminal or by imbedding them in the text. The set command contains interface initialization and character output routines to support the SWTPC MP-C interface as well as the standard serial and parallel interfaces. Jumpers are also provided to user supplied printer routines. User selects the port address (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routines. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reenter (Warm Start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no echo) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 or 6809, SSB or FLEX™, 5" or 8"	45.00
Printed source listing is available for an additional	35.00
Software by Technical Systems Consultants, Inc.	
Flex™ (includes Editor and Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	450.00
Editor	50.00
Assembler	50.00
6809 Cross Assembler on 6800	100.00
68000 Cross Assembler on 6800 or 6809	250.00
Text Processor	75.00
Basrc	75.00
Extended Basic	100.00
Basic Precompiler (specify standard or extended)	50.00
Multi-User Basic for S/09	150.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™)***	225.00
***One year maintenance and update	add 75.00
6800 Flex™ Utilities	100.00
6809 Flex™ Utilities	75.00
Debug Package	75.00
Diagnostic Package	75.00
The following are available for 6800 only	
Soap IV (stack oriented arithmetic processor)	25.00
Floating Point Package	25.00
Scientific Functions Package	25.00
Relocator	25.00
Disassembler	25.00
Micro Basic Plus	25.00
Space Voyage	25.00
6800 Game Package III	25.00
6800 Mini-Monitor	5.00
Software by Microware Systems Corp.	
OS-9™ Level One Operating System**	195.00
OS-9™ Level Two operating System**	495.00
BASIC09***	195.00
**Yearly Maintenance and Update	add $75.00
OS-9™ Text Editor	95.00
OS-9™ Interactive Assembler	95.00
OS-9™ Interactive Debugger (Disk version)	35.00
RT/68 Real Time Operating System (6830 or 2708)	75.00
Specify manufacturer and type of CPU and I/O controller	
ABASIC on cassette (Limited quantity)	Phone for special price
SWTPC	
8209 9" Intelligent Terminal	945.00
8212 12" Intelligent Terminal	995.00
8212W 12" Intelligent Terminal with word processing enhancements	1095.00
DC-1 or DC-2 5" Disk controller (limited quantity)	125.00
CT-64 Terminal kit in original factory sealed box (limited quantity)	326.00
DMF-2 8" DMA double density, double sided controller board	495.00
MP-S Serial interface (single port, kit)	40.00
MP-S2 Serial interface (dual port, assembled)	120.00
MP-LA Parallel interface (dual port, kit)	40.00
MP-L2 Parallel interface (dual port, assembled)	120.00
MP-R Single voltage 2716 prom programmer (kit)	49.95
MP-R Single voltage 2716 prom programmer (assembled)	99.50
MP-N Calculator board (kit)	54.95
MP-N Calculator board (assembled)	80.00
MP-T Interrupt timer (kit)	47.50
MP-T Interrupt timer (assembled)	80.00
MP-8M 8K 4044 Memory board (kit, Special, limited quantity)	180.00
MP-8M 8K 4044 Memory board (assembled)	235.00
S32 Universal Static Memory Board	99.50
S3216 Universal Static Memory Board with 16K Ram	295.00
S3232 Universal Static Memory Board with 32K Ram	495.00
MP-09 6809 CPU board kit (add $50.00 for assembled with sockets)	175.00
69A Chassis, P.S., 2MHz 6809 Processor, 8K. RAM, One Serial Port	695.00
68/K Kit version of above	575.00
6800 CPU and Disk Controller Boards	
Due to the relative unavailability of these SWTPC items, please refer to the next column for ELEKTRA and GIMIX alternatives. Phone for SWTPC availability.	
Universal 68XX Motherboard. SS-50/50C, 4/16 addresses per port	60.00
Connectors (10 pin, Titanium-Tin plated 5 microns for near gold quality)	
Male with square cross section pins	each .50
Female	each .75

SMOOTH and **ELEKTRA** are trademarks of AAA Chicago Computer Center.
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.

ELEKTRA™ SS50 Computer Products

OPSb Dual Port Serial Interface bare board and documentation	20.00
OPPb Dual Port Parallel Interface bare board and documentation	20.00
MBb Motherboard bareboard and documentation* (motherboard is terminated and has ground lines separating each of the data and address lines)	
6/9b 6808-6809 CPU bare board and documentation* (6808 supports 6800 software)	
*Price to be announced when product is available.	
HUMBUG (from STAR-KITS) for 6/9 CPU board	
2K version for 6800 (6808)	40.00
HUMBUG-09 for 6809	75.00
Other HUMBUG versions including video versions are available. (Specify system)	
Cabinet (heavy gauge, 2 cutouts for 5 1/4" disk drives, drive mount, line cord, line fuse, power switch, reset switch, 70 cfm fan, EMI filter, 8 RS-232 cutouts) and power supply (20a 8v, 4a 16v, 4a -16v), use any current SS-50 or SS-50C motherboard	395.00
Cabinet without power supply	250.00
Power supply (20a 8v, 4a 16v, 4a -16v)	175.00
5" disk regulator board	25.00
Filler plate for 5-1/4" cutout	10.00
MPI 5 1/4" DISK DRIVES with 30 day guarantee and 5 msec stepping rate	
Our service department thoroughly tests every MPI drive before we ship.	
B51 - Single head, single or double density capability, 40 tracks	$250.00
B52 - Double head, single or double density capability, 40 X 2 tracks	335.00
B91 - Single head, single or double density capability, 80 tracks	335.00
B92 - Double head, single or double density capability, 80 X 2 tracks	470.00
MPI - Service Manual	20.00
Alignment Service (5 1/4" or 8" floppy drives)	60.00
Repair Service (Cost based on parts and labor)	
Minimum fee if no trouble found	25.00
Dual drive cabinet for 5 1/4" drives with Power supply, line cord, fuse, power switch, and Power cables to drives	125.00
Dynamite Disassembler	60.00
SUPER SLEUTH Disassembler System	99.00
Microtime 6800 Calendar and Clock Board (assembled and tested)	105.00
Bareboard, connector, and documentation only of above (See review Feb. 1980 '68' Micro Journal)	35.00
Epson Printer (Centronics compatible parallel interface)	495.00
(with Serial RS 232 interface option)	add 75.00
Spare Print Head	30.00
Spare ribbon cartridge	15.00
Optimal Technology, Inc. EP-2A-79 Eprom Programmer	169.00
PM-0, PM-1, PM-3, PM-5, PM-7 Personality Modules	each 18.00
PM-2, PM-4	each 34.00
PM-3	26.00
PM-8	36.00
GIMIX (The Ultimate)	
6800 CPU Board	224.03
with timers	288.06
with baud rate option	add 30.00
with 2MHz option	add 15.00
2 MHz 6809 Plus CPU, time of day clock, battery backup, 1K NMOS RAM	578.05
CMOS RAM substitution	8.00
GIMIX Dynamic Address Translator	35.00
SWPTC compatible DAT	15.00
9511A Arithmetic Processor (4MHz)	312.00
9512 Arithmetic Processor (3MHz)	265.00
GMXBUG-09 (Terminal Based) 1K scratchpad required	98.65
Bootstrap Prom	30.00
Video Prom (includes bootstrap)	30.00
Manual and Source Listing only	38.62
Missing cycle detect card	36.23
Disk Controllers (All have data separators and can be used with either single or double headed drives)	
5" single density controller without 1771 chip	158.38
5" single density controller complete	198.48
5" and 8" single density controller complete	226.58
5" double density controller with variable precomp	348.28
DMA 5" AND 8" double density controller with variable precomp	548.68
GIMIX version of FLEX™ (without Editor and Assembler)	90.00
Double disk regulator card	68.22
Ribbon cable for two 5 1/4" disk drives (short)	34.96
Ribbon cable for two 5 1/4" disk drives (long)	39.96
Ribbon cable for two 8" disk drives (long)	44.26
Memory	
32K Static RAM Board with 16K of RAM installed	298.12
32K Static RAM Board with 24K of RAM installed	348.14
32K Static RAM Board with 32K of RAM installed	398.15
64K Static RAM Board with 56K of RAM installed	994.56
64K Static RAM Board with 64K of RAM installed	10.8664
16 Socket EPROM/ROM/RAM Board	238.32
8K Promboard (2708)	98.34
4K PPO 4K Prom Board and 2708 Prom Burner	198.00
I/O Boards	
Single port 30 Pin serial interface (Requires 1 cable set)	88.41
Dual port 30 pin serial interface (Requires 2 cable sets)	128.43
8 port 50 pin serial interface with baud rate generator	318.46
Dual port 30 pin parallel interface (Requires 2 cable sets)	88.42
8 port 50 pin parallel interface with interrupt generator	198.45
Cable sets for above boards (specify board)	22.95
Video Boards	
64 or 32 X 16	198.71
80 X 24 without RAM character generator	398.74
80 X 24 with RAM character generator	458.76
High resolution (512 X 512 dot resolution)	998.77
2MHz 6809 PLUS Computer System with 56K Memory	2498.29
Mainframe (Chassis, PS, Switches, Fan, Motherboard, Baud Rate Gen.)	998.19

Prices subject to change without advanced notice

Phone Consultation:

Most weekdays	4 PM to 6 PM
Saturdays (except July and Aug)	11 AM to 5 PM

Shipping and handling estimates:

Within the Continental U.S., please add $5.00 or 3% whichever is greater.
Foreign, prepaid and add $10.00 or 10% whichever is greater (light items only). Heavy items must be prepaid and will be shipped Emery Air Freight Collect.
Please phone during consultation hours if questions arise regarding shipping fees.

Master Charge, Visa, and American Express honored

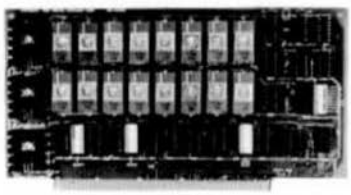
AAA Chicago Computer Center
120 Chestnut Lane, Wheeling, IL 60090
(312) 459-0450

SEE GIMIX AD PAGES 3 & 44

Dealer for GIMIX, SSB, SWTPC, Microware Systems Corp., and Technical Systems Consultants, Inc.

DIGITAL RESEARCH COMPUTERS
(214) 271-3538

32K S-100 EPROM CARD
NEW!



$79.95 KIT

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $8.95 Ea. With Above Kit

KIT FEATURES:
1. Uses +5V only 2716 (2Kx8) EPROM's.
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board, solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

32K SS-50 RAM

$329.00 KIT

For 2MHZ Add $10

Blank PC Board $50

For SWTPC 6800 - 6809 Buss

Support IC's and Caps $19.95

Complete Socket Set $21.00

Fully Assembled, Tested, Burned In Add $30

NEW!

At Last! An affordable 32K Static RAM with full 6809 Capability.

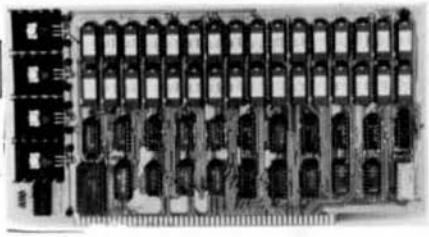
FEATURES:
1. Uses proven low power 2114 Static RAMS.
2. Supports SS50C - EXTENDED ADDRESSING.
3. All parts and sockets included.
4. Dip Switch address select as a 32K block.
5. Extended addressing can be disabled.
6. Works with all existing 6800 SS50 systems.
7. Fully bypassed. PC Board is double sided, plated thru, with silk screen.

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

$169.95 KIT

FOR 4MHZ ADD $10



KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-$33

LOW PROFILE SOCKET SET-$12

SUPPORT IC'S & CAPS-$19.95

ASSEMBLED & TESTED-ADD $35

OUR #1 SELLING RAM BOARD!

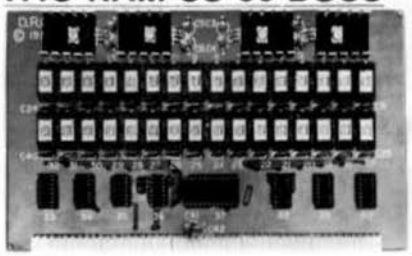
16K STATIC RAM SS-50 BUSS

PRICE CUT!

$159 KIT

FULLY STATIC!

FOR 2MHZ ADD $10



FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES:
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout.
5. All Parts and Sockets included
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—$35 COMPLETE SOCKET SET—$12
SUPPORT IC'S AND CAPS—$19.95

NEW! **STEREO!** NEW!
S-100 SOUND COMPUTER BOARD

At last, an S-100 Board that unleashes the full power of two undetectable General Instrument AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be called in ASIC, ASSEMBLY LANGUAGE, etc.

KIT FEATURES:
- TWO GI SOUND COMPUTER IC'S.
- FOUR PARALLEL I/O PORTS ON BOARD.
- USES ON BOARD AUDIO AMPSOR YOUR STEREO.
- ON BOARD PROTO TYPING AREA.
- ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
- PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
- EASY, QUICK, AND FUN TO BUILD, WITH FULL INSTRUCTIONS.
- USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY.

Both Basic and Assembly Language Programming examples are included

COMPLETE KIT! **$84.95** (WITH DATA MANUAL)

BLANK PC BOARD W/OATA $31

SOFTWARE:

SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register-Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M* compatible diskette or 2708 or 2716 Diskette - $24.95 2708 - $18.95 2716 - $29.95. Diskette includes the source. EPROM'S are ORG at E000H.

SPECIAL PURCHASE!

UART SALE!

TR1602B — SAME AS TMS6011, AY5-1013, ETC. 40 PIN DIP

TR1602B

$2.95 EACH **4 For $10.00**

CRT CONTROLLER CHIP

SMC #CRT 5037, PROGRAMMABLE FOR 80 x 24, ETC. VERY RARE SURPLUS FIND. WITH PIN OUT. $12.95 EACH.

4K STATIC RAM

National Semi, MM5257, Arranged 4K x 1, +5V, 18 PIN DIP, A Lower Power, Plug in Replacement for TMS 4044, 450 NS. Several Boards on the Market Will Accept These Rams. SUPER SURPLUS PURCHASE! PRIME NEW UNITS!

8 FOR $16 32 FOR $59.95

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control, 16 bit Envelope Period Control, 2-8 Bit Parallel I/O, 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy interface to the S-100 or other busses. $11.95 PRICE CU

SPECIAL OFFER: ~~$14.95~~ each Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

*TRADEMARK OF DIGITAL RESEARCH.

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWAI

DATA BASE MANAGER

NOW AVAILABLE . . . .

Accts. Receivable/Order Input	$495.00
Accts. Payable/Purchase Order	$495.00
General Ledger	$495.00
Manufacturer's Inventory	$495.00
Payroll	$495.00
Scheduled Maintenance Program	$495.00
Inventory II	$495.00
Manuals with Print-Outs	$ 20.00

DBM 2 FLEX* . . . .$450.

UNIFLEX* . . . . .$550.

Custom Programming Services Available

DBM 2 and DBM UNIFLEX contain source listings.

Create Data Files
Build Files
Edit Files
Sort Programs
Generate Report Programs
File Utility Programs
General Utility Programs

SUPER ENHANCED DBM II

SOFTWARE AVAILABLE THROUGH THESE DEALERS

Barney & Associates 113 W. Fifth Pittsburg, Kansas
Bonelli Media Assoc. 1116 N. 38th St. Allentown, Pa. 18104
Data Comp. P.O. Box 794 Hixson, Tn. 37343
Data Flex 7 Beaufort Ave. Craighall Park Johannesburg, 2196 South Africa
Digital Services 1509 Tieton Dr. Yakima, Wa. 98902
MRLectronics B.V. Buitendofdreef 280 2625 Re delft. The Netherlands
New Technology Systems P.O. Box 2 Canatlan, DGO Mexico
Paris Radio 7A Burton St. Darlinghurst,NSW Australia
SWTP 38 Dover St. London, England
The Computer Store P.O Box 4045 Little Rock, Ark. 72214

2457 Wehrle Drive • Buffalo, New York 14221 • (716) 631-3011

*FLEX AND UNIFLEX ARE TRADEMARKS OF TECHNICAL SYSTEMS CONSULTANTS.

UNIVERSAL DATA RESEARCH INC.

UNIFACE BR-1 BIT RATE GENERATOR

- ☐ Supplies standard baud signals or jumper selectable high baud option
- ☐ Facilitates upgrading to S-50C extended addressing
- ☐ Uses chips and crystal from your 6800 processor board without losing processor function (SWTPC MPA requires Upgrade Kit)

Complete and tested	$65.00
Assembled and socketed (without chips & crystal)	$44.00
MPA Upgrade Kit	$10.00
Documentation only (credit to future purchase)	$ 5.00

UNICOMP BT-1 ACTIVE BUS TERMINATOR

- ☐ Presents 120 ohm active termination
- ☐ Allows greater than 5 megahertz operation on all standard motherboards
- ☐ Can eliminate critical timing and noise problems when using fast devices (e.g. dynamic memories, hard disk drives, etc.)

Assembled and tested $75.00

Gold plated connectors are standard on all Uniface S-30 and Unicomp S-50 products

microdyne Add $3.00 Shipping

P.O. Box 1707 • Greenville, MS 38701 • (601) 335-9321

ADVENTURE GAMES

6800 / 6809

- ALL NEW, EXCITING SERIES
- CHALLENGING SITUATIONS
- FAST, EFFICIENT MACHINE LANGUAGE
- REQUIRES 16K OF MEMORY PLUS DOS
- SAVE GAME IN PROGRESS
- 5¼" DISC, TSC FLEX COMPATIBLE

MARK DATA PRODUCTS IS PRODUCING A NEW SERIES OF ADVENTURE GAMES FOR YOUR 68XX.

CALIXTO ISLAND - 24.95
NEW - BLACK SANCTUM -24.95 - NEW
ORDER BOTH GAMES FOR 39.95

MORE COMING SOON

ATTRACTIVE DEALER DISCOUNTS

SHIPPED PREPAID IN CONTINENTAL U.S.
CALIFORNIA RESIDENTS, PLEASE ADD 6% TAX

TRS80 COLOR VERSIONS 19.95 EACH

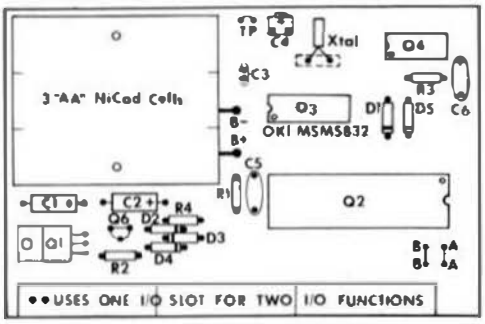
MARK DATA PRODUCTS

23802 BARQUILLA MISSION VIEJO, CALIFORNIA, 92691

(714) 768-1551 - EVENINGS

FLEX IS A TRADEMARK OF TSC, INC.
TRS80 IS A TRADEMARK OF TANDY CORP.

Model 6800CL4 CalClock/TIMER



IT'S A HARDWARE CALENDAR/CLOCK

- Keeps date and time without servicing by the computer
- Day-o-week, month/day/year, hour/min/sec (12/24hr.+ auto Leap Year)
- Hands off setting/control/access of ALL functions via software
- On-card battery and charging circuit keeps time for months power off

WITH AN INTERVAL TIMER INCLUDED

- For (TSC/Flex 2/9® compatible) printer spooling, multi-tasking, etc.

ully assembled & tested •	$ 99.95	5" Disk (Flex2☐ Flex9☐)®	$ 10.00
Complete kit®	$ 69.95	Goldplated buss connectors	$ 6.00
Bare board®	$ 35.00	Shipping & handling	$ 3.00

• FULLY DOCUMENTED: instructions; diagrams; theory; more than 20 pages of sample software (automatically puts date in Flex2/9® date buffer, adds time-of-day to assembly listings, maintains constant current time+date display on top line of CRT). Batteries not included. All IC's socketed.

® FLEX is the registered trademark of Technical Systems Consultants, Inc.

COMPUWARE Corporation
P.O. Box 2710
Cherry Hill, NJ 08003
609-428-2309

Dealer and Volume Discounts Available

New Jersey buyers ADD 5%
Terms: CASH; MC; or Visa
Flex9®☐ Flex2®(default)☐

$50.00 EZDATA $50.00
MANAGEMENT SYSTEM

EASY TO CREATE
EASY TO MAINTAIN
EASY TO REPORT

True random access (FAST)	Absolute Cursor Control
Max. records on file (65,535)	Insert A Character/Expand Field
Runs in under 16K	Delete A Character/Contract Field
6800 assembly code (FAST)	Screen Paging - 126 page Max.
Full record/Screen updating	Online/Real Time updating
Change File sequence on demand	Full Forward/Backward/ Search Control
Sort/Print any item in any Sequence	Cursor stops ONLY on the item to be Added/Changed

NOW . . . 6809 COMPATIBLE . . . NOW

Interact with your data in a language we can all understand
— ENGLISH —

All commented source Programs on
Three (3) 5 1/4" or One (1) 8" Disc.

Included: EZCTL, EZDATA, EZTEXT, EZVIEW, EZEDIT, EZPRINT, EZPURG, EZSORT

Please specify SSB, FLEX 09, FLEX & 5¼" or 8" Disk

GARY A. MAGNUSEN
208 TINKLER
LAFAYETTE, INDIANA 47901

Evenings 1-317-742-1565

Flex is a Trademark of Techinical Systems Consultants.

If you need M68000, Z8000, or 8086 Software, take it off our shelf.

Why wait up to 6 months for high quality 16-bit resident system software, when you can get it now?

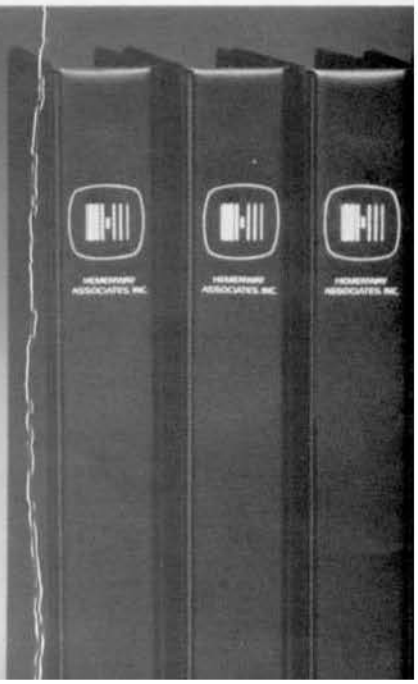
Hemenway Associates can give you the software tools you need to get the most out of your 16-bit hardware. Because 8086, Z8000 and 68000 Systems Software is sitting on our shelves right now.

Single user operating systems, macro assemblers, text editors, floating point packages, multi-tasking operating systems, linking loaders and PASCAL compilers are all ready for immediate delivery.

You've heard about Hemenway Associates' system designs in the pages of EDN. But now you can have Hemenway-designed software for your 16-bit microprocessor. Our software is easily adaptable to any hardware configuration. From manufacturers evaluation boards to full-blown microcomputers. It's efficient, easy-to-use and backed by ongoing support.

So whether you're a microcomputer manufacturer, software developer or individual microcomputer owner, Hemenway has the software for you.

For the hard facts on Hemenway software, write or call: Hemenway Associates, 101 Tremont Street, Boston, Massachusetts 02108. Phone: 617-426-1931. TWX 710321 1203. TELEX 921735.



HEMENWAY ASSOCIATES, INC.

When it comes to software, come to Hemenway

NEW REDUCED PRICES!

SOUTHEASTERN MICRO SYSTEMS inc.

404-922-1620

NEW PRODUCT!

DS-16

SERIAL INTERFACE

- TWO ACIA PORTS
- TWO ON BOARD DB-25 CONNECTORS (on top of board)
- HAS ON BOARD BAUD RATE GEN
- BAUD RATE MAY BE PROGRAMMED FOR ON OR OFF BOARD USE
- BOARD CAN USE EXTERNAL BAUD RATE FROM BUSS OR RS-232
- BAUD RATE IS PROGRAMMABLE TO 32K BAUD
- FOR THE SS-30 BUSS
- ALL LINES BUFFERED
- ALL REGULATORS 1 AMP. +5 +12 -12

ASSEMBLED	BARE
118.88	$30

8M-58

STATIC RAM

- 2 MHZ OPERATION
- LOW POWER (2 amp)
- EXTENDED ADDRESSING **56K**
- ALL I/O LINES BUFFERED
- ADDRESSABLE IN 8K BOUNDRIES
- DESIGNED FOR 8K x 8 STATIC CHIPS

ASSEMBLED	BARE
$550	$75

DDC-18

DISK CONTROLLER BOARD

- CAN BE PROGRAMMED FOR COMPATIBILITY WITH SWTP DC4
- WRITE PRECOMP/HARDWARE OR SOFTWARE SELECTABLE
- WRITE PROTECT HARDWARE OR SOFTWARE SELECTABLE
- SIDE SELECT SOFTWARE - SWTP/OTHER
- DS/DD
- ON BOARD DATA SEPARATOR (1691 & 4 phase clock)
- 4 OR 6 ADDRESS PORT
- WILL RUN SS/SD TO DS/DD

ASSEMBLED	BARE
$250	$50

BTR-09

TERMINATOR BOARD

- FOR SS-50 AND SS-50C BUSS
- WILL REMOVE MOST ADDRESS AND DATA LINE NOISES
- SYSTEMS RUN WITH LESS ERRORS
- DISK SYSTEMS PERFORM BETTER

ASSEMBLED
$19.75

DM-64

MEMORY BOARD

- USES LOW COST 8716 DYNAMIC RAM
- TOTAL TRANSPARENT REFRESH: OPERATES LIKE STATIC
- LOW POWER CONSUMPTION **Bare**
- ADDRESSABLE IN 8-K BLOCKS **$75**
- CAN BE USED WITH 16K TO 64K
- EXTENDED ADDRESSING TO 16 BOARDS
- OPERATES ON SS-50 AND SS-50C BUSS
- SOLDER MASK AND SILKSCREENED BOARDS
- ALL MEMORY CHIPS ARE PRIME 8116-200ns

KITS	ASSEMBLED
16K Ram $249.00	16K Ram $303.00
64K Ram $375.00	64K Ram $429.00

RR-32

STATIC RAM/ROM BOARD

- 2016 (2K x 8 Static)
- 2716 (2K x 8 Eprom)
- 2732 (4K x 8 Eprom) **32K**
- EXTENDED ADDRESSING
- ADDRESSABLE IN ANY 2K BOUNDRIES
- MIX RAM & ROM CHIPS ON SINGLE BOARD
- LOW POWER - LESS THAN 2 AMPS FOR 56K
- ALL IO LINES ARE BUFFERED
- DESIGNED FOR 2 MHZ OPERATION

ASSEMBLED	BARE
$375	$75

ST.02

VIDEO BOARD

- STAND ALONE VIDEO BOARD
- BAUD RATES 300-9600
- SCREEN PRINTER PORT
- STANDARD RS-232 INTERFACE
- 8 BIT KEY BOARD PORT
- PROGRAMMABLE SCREEN FORMAT

ASSEMBLED	BARE
395	$75

ADD $4.00 FOR SHIPPING IN USA; 5% FOREIGN (US FUNDS ONLY)

P.O. BOX 295
1080 IRIS DRIVE. SUITE 20E
CONYERS, GA 30207

THOMAS INSTRUMENTATION

THE MACHINE TOOL, INDUSTRIAL SPECIALISTS IN BUSINESS ON A FULL TIME BASIS FOR 10 YEARS

NEW PRODUCT: **S-R/R**

48K 2MHz STATIC RAM/ROM CARD

- *24 2K blocks memory mapped on any 2K boundary
- *uses low power 2016P-2 (2128) RAM and/or 2716 ROM
- *mix 4K blocks of RAM and ROM
- *6800 and 6809 compatible
- *use on SS-50 and SS-50C buss
- *decoded for extended addressing
- *5 volts only
- *low power consumption (typ. ½ amp with 48K RAM)
- *gold connectors

Special Introductory Prices

Bare board $39.20 2716 $12.80 2016P-2 $13.60

A/T without memory chips $100.00

A/T with 16K $220.00; with 32K $316.00; with 48K $440.00

NEW ACCESSORIES FOR 68XX USERS:

SS-50/SS-50C EXTENDER CARD	$35.00
SS-30 EXTENDER CARD	$25.00
*Both cards assembled with a built in logic aid & gold edge connectors	
SS-30 WIRE-WRAP/PROTOTYPE BOARD (board only)	$20.00
*Pad spacing permits most standard sockets from 8 to 64 pins	
*Provision has been made for voltage regulators	

FEATURED PRODUCT: **SP-1 Bare card $49.00 Asm. + tested $195.00**

*A super prototype board *Card design includes

(3) 6821 6 parallel ports
(4) 6850 4 serial ports
(1) 6840 3 16 bit counter/timers
which are fully buffered and decoded

*Accomodates a mix of 38, 14 & 16 pin wire wrap sockets
*Pad spacing permits most standard sockets from 8 to 64 pins

MODEM CARD	B/C	$ 49.00
special parts kit		$195.00
A/T without extra features		$325.00
*SUPER CPU assembled with source listing without 2K EPROMS (2-2708)		$235.00
*Monitor in two 2708 EPROMS		$ 29.00
*CPU bare card, doc., & src.		$ 59.00
*VIDEO RAM asm. 7x9 chars 64x16		$195.00
*VIDEO RAM bare, doc, Xtal, src.		$ 49.00
*PARRALLEL I/O asm 100 I/O lines incl. 5 PIA's for 10 ports		$139.00
*PARALLEL I/O bare card & doc.		$ 49.00
*SS-50 WIRE-WRAP/PROTOTYPE bare		$ 39.00
*TRANSITION CARD asm.		$ 95.00
*TRANSITION CARD bare		$ 49.00

A/T with extra features	$395.00
Software obj. & src. on FLEX disk	$ 10.00
BACKPLANES AND MOTHERBOARDS	
*16 position SS-50	$80.00
*12 position SS-50	$60.00
* 8 position SS-50	$40.00
* 6 position SS-50	$30.00
* 4 position SS-50	$20.00
* 8 position SS-30	$39.00

**Connectors:
GOLD $1.60 ea. (M or F)
TIN M $.40 ea. F $.50 ea.

DEALERS FOR SWTPC, GIMIX, AND TSC

*All Thomas Instrumentation's cards come with full documentation including software source listings where applicable *All assembled cards are burned in at 150F and fully tested with Gold conn. *Bare card prices do not include edge connectors *See previous ads, write, or call for more detailed information.

THOMAS INSTRUMENTATION

168 EIGHTH STREET — AVALON, N.J. 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
CONT. USA INCLUDE $3.00 SHIPPING, CANADA $6.00, FOREIGN $12.00
MASTERCARD, VISA, and C.O.D. ACCEPTED
To satisfy in-depth questions regarding our products send $20.00 to receive full documentation, schematics, & source listings for all boards currently in production

64K BYTE CMOS STATIC RAM BOARD . . . with Battery Back-Up

Using the latest in memory technology, the GIMIX 64K BYTE CMOS STATIC RAM BOARD combines the best features of previous memory boards on one board.

FULLY STATIC MEMORY with its inherent low soft error rate and freedom from alpha-particle induced errors. No complicated refresh timing or clocks required for data retention. Fully compatible with any of the 6800/6809 DMA techniques.

HIGH SPEED 200ns. memorys for guaranteed operation at 2MHz. with no wait states or clock stretching required.

ULTRA-LOW POWER CMOS RAM requires less than 1/4 AMP (250 Ma.) at 8V. for a fully populated 64K BYTE board. Less power supply loading and heat generation for cool, efficient operation.

NON-VOLATILE using an on-board nickel-cadmium battery. The board retains data even with system power removed. With the battery fully charged, the contents of the memory remain intact for a minimum of 21 days.

HIGH DENSITY permits greater memory expansion to meet the needs of todays sophisticated, multi-user/multi-tasking operating systems.

ADDRESSABLE in two 32K sections that have their own decoding for both the regular and extended (SS-50C) address lines. Each section can be addressed to any 32K boundary in the address range (1M BYTE with extended addressing). The 32K sections are divided into four 8K blocks that can be individually enabled or disabled. Disabled sections do not occupy any address space.

RELIABLE like all GIMIX products, the 64K BYTE CMOS STATIC RAM is designed with reliability in mind. Series damping resistors, a fully gridded power and ground layout, and generous power supply decoupling, all contribute to reliability and data integrity. An unsafe voltage detect circuit inhibits writes to the board, when the 8V. supply falls below a preset level, to prevent loss of data during the transition between system and battery power.

The GIMIX 64K BYTE STATIC RAM BOARD is ideally suited to a wide variety of applications.

Its high density and ultra-low power consumption make it possible to greatly expand systems with a few available bus slots and limited power supply capabilities.

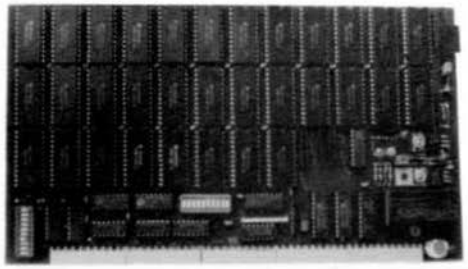
The battery back-up feature is useful where data loss due to power failure cannot be tolerated, or as a replacement for disk or tape storage where conditions such as environment prohibit their use. Since the entire board can be hardware write protected by a switch located at the top of the board, it can also be used to emulate PROM or ROM memory. This is especially useful during firmware development where frequent software changes must be made.

When the board is used in conjunction with a device such as the GIMIX MISSING CYCLE DETECTOR BOARD, which monitors the A.C. line and generates an interrupt when a power failure occurs, critical data can be stored and system integrity maintained during either expected or unexpected power outages.

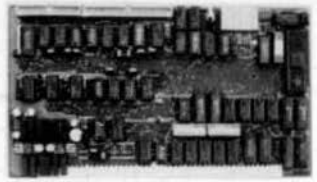
The **GIMIX 64K BYTE STATIC MEMORY BOARD** is available in 56K and 64K versions. Both version include all of the above features; gold bus connectors; and come fully assembled, burned in, and tested.

56K version . . . **$ 994.56**
(Socketed for 64K)

64K version . . . **$1088.64**



GIMIX KNOCKS OUT DISK PROBLEMS



GIMIX DMA DOUBLE DENSITY DISK CONTROLLER #68

The GIMIX DMA (Direct Memory Access) DISK CONTROLLER has the capabilities needed to realize the full potential of todays sophisticated multi-user/multi-tasking operating systems such as OS-9™ and UniFLEX™.

HIGH SPEED using bi-polar logic DMA circuitry for guaranteed operation at 2MHz. DMA transfers take place at full bus speed using 6809 cycle steal DMA. Once the required parameters are passed to the controller and DMA transfer is initiated the processor is free for other tasks. Interrupts can be generated to indicate the completion of the transfer.

SINGLE AND DOUBLE DENSITY data storage on any combination of 5¼" and 8" floppy disk drives, single and double headed, single and double track density, up to 4 drives total.

LOW ERROR RATES are insured by a phase lock data recovery circuit (data separator) and adjustable write precompensation circuitry for drives that require precomp. Separate precomp adjustments are provided for 5¼" and 8" drives.

ADDRESSABLE to any 8 byte boundary in the address space (1M byte when extended address decoding is used). The board occupies only 8 bytes of address space.

EXTENDED ADDRESSING control using the SS-50C extended address lines. Control of the extended address lines allows the board to perform DMA transfers to and from any address in the 1M byte address space.

FULLY BUFFERED with separate 5¼" and 8" output buffers and schmitt trigger input buffers for the disk drive signals.

The DMA controller frees the processor free to perform other tasks once the transfer is initiated, while programmed I/O disk controllers which require full time use of the processor during data transfers to and from disk.

This is extremely important in a multi-user / multi-tasking environment as the processor can perform other tasks such as console I/O while a disk transfer is in progress.

#68 fully assembled, burned in, and tested **$548.68**

GIMIX DOUBLE DENSITY PIO DISK CONTROLLER #28

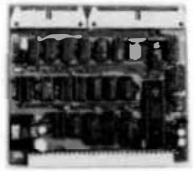
The GIMIX DOUBLE DENSITY PIO (PROGRAMMED I/O) DISK CONTROLLER is a versatile floppy disk interface for use on 6809 systems on the SS-50 or SS-50C bus. The board physically occupies one slot of the 30 pin I/O bus.

- Double the unformatted storage capacity of single density controllers
- Single and double density operation
- Phase lock data recovery circuit (data separator)
- Adjustable write precompensation circuitry
- Controls up to four 5¼" drives
- Controls single and double headed drives
- Designed to meet the data hold-time requirements of the Western Digital 1797 floppy disk controller I.C.

The GIMIX DOUBLE DENSITY PIO DISK CONTROLLER is ideal for systems that require greater disk storage than that provided by single density controllers, without increasing the number or type of drives. In most cases existing 6809 systems can be upgraded by adding only the controller and the appropriate operating system disks are.

#28 fully assembled, burned in and tested **$348.28**

GIMIX 5/8 DISK CONTROLLER BOARD #58



The GIMIX 5/8 DISK CONTROLLER #58 is a versatile floppy disk interface for use with both 6800 and 6809 systems on the SS-50 or SS-50C bus. The board physically occupies one slot of the 30 pin I/O bus.

- Hardware and software compatible with existing disk controllers (SWTPc DC-1, DC-2 and DC-3)
- Controls up to four 5¼" drives on 6800 systems
- Controls any mix of 5¼" and 8" drives, up to four drives total, on 6809 systems
- Provides for double headed drives
- Synchronized data separator for data reliability
- Designed to meet the data hold-time requirements of the 1771 floppy disk controller I.C.

The GIMIX 5/8 DISK CONTROLLER is ideal for a variety of applications including the conversion of computers or existing systems. As a replacement it can provide the added advantages of a data separator, double headed drive capability, and a 6809 system the ability to use 8" drives. Double headed drives and 8" operation may require appropriate operating system software.

#58 fully assembled, burned in, and tested **$226.58**

NOTE: When ordering disk controllers please specify the make and model of the drives being used.

GIMIX 6809 FLEX™

GIMIX versions of Technical Systems Consultants 6809 FLEX™ operating system are available for all three GIMIX disk controllers. They fully support all the features of each controller and are software compatible with other versions of FLEX™. GIMIX FLEX™ includes a disk FORMAT program that allows the user to pick the number of tracks to format, single or double sided disks, and where appropriate single or double density. GIMIX FLEX™ supports single and double track density (48 and 96 TPI 5¼" drives) and allows 96 TPI (80 track) drives to read, write, or format 48 TPI (35 or 40 track) disks.

GIMIX 6809 FLEX™ specify controller and type of drive
8", or 5¼" 40 track (48 TPI), or 5¼" 80 track (96 TPI) **$90.00**

Versions of MICROWARE'S OS-9™ will be available for all three GIMIX disk controllers. Technical Systems Consultants' Uni-FLEX™ will be available for the GIMIX DMA controller.

GIMIX INC.
1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510 • TWX 910-221-4055

Epson Factory Authorized Repair Center

SOUTH EAST MEDIA

P.O. Box 794 Chattanooga TN 37443

1-615-877-2241

10 CPI Standard

ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRST

abcdefghijklmnopq

1234567890123456

EPSON MAKES MORE

Epson makes more

10 CPI Emphasized

ABCDEFGHIJKLMNOPQ

abcdefghijklmnopq

1234567890123456

EPSON MAKES MORE

Epson makes more

5 CPI Double Emphasized

ABCDEFGH

abcdefgh

12345678

EPSON MA

Epson ma

IN THE WORLD.

in the World.

IN THE WORLD.

in the world.

CHANISMS

chanisms

For most people, spending money for a printer with capabilities they'll never use just doesn't make sense. That's why we developed the MX-70.

The MX-70 is the lowest priced 80-column dot matrix printer on the market today. But it's still one fine little printer. In fact, the MX-70 has features that until recently you couldn't buy at any price.

One of the family.

The MX-70 is one of the Epson MX Series of printers. It shares technology based on years of intensive research and development. Its patented, disposable print head prints between 50 and 100 million characters. When it wears out, just throw it away. It's user replaceable at about 1/3 the cost of standard print heads.

Our precision plastic injection molded case eliminates most costly hand assembly, which keeps the price low and the quality high.

Everything you need.

The MX-70 prints at 80 CPS with a user-defined choice of 80 adjustable-width character level or 80-column printing. It gives you top-of

forms recognition, programmable line feed and form lengths, an easy-to-read 5x7 matrix, self test, and an adjustable tractor feed. And because it's a true impact printer, it doesn't need exotic aluminized or thermal paper; plain paper is just fine.

And we give you graphics at no extra charge.

GRAFTRAX II is one of the finest bit-plot graphics capabilities on the market today. And on the MX-70 it's free. With unparalleled dot placement accuracy, the bit-image graphics are true to the letter, and with the plastic platen, much other printers. So you get blacker blacks, cleaner grays and definition you just won't find anywhere else. That's not a bad bargain on a low-cost printer, eh?

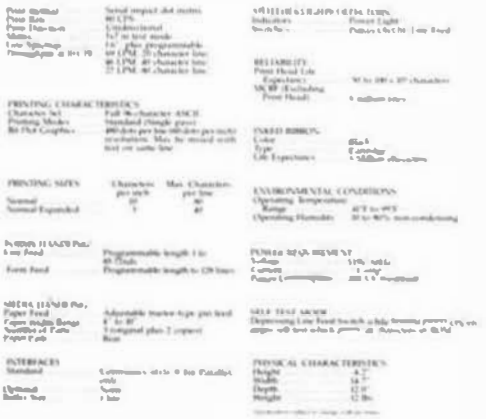
No-frills printing at a no-frills price.

If you want a simple, no-frills printer that you don't want to buy features you'll never use, if you want a reliable printer backed by the world's largest manufacturer of print mechanisms, we're ultra-there's only one logical choice.

The Epson MX-70

The Epson MX-70

$375.00



The Epson MX-80

$495.00

This is a South East Media Special to 68 Micro Journal Readers.

Add $7.50 Shipping and Handling, Allow stock to 3 weeks.

COMING SOON MX-80FT AND THE MX-100

'68' Micro Journal
3018 Hamill Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

THE SOFTWARE FOR THE HARDCORE

tFORTH™ THE PROFESSIONAL'S CHOICE

WHAT? FORTH! WHO? TALBOT!

Extended version of the original 6809 implementation of FORTH INTEREST GROUP FORTH by astrophysicist Dr. Ray Talbot; equivalent 6800 version also available. Now obtain tFORTH direct from the author, thereby ensuring direct and quick support. Supplied on FLEX™ soft-sectored diskettes, although when you discover the elegant simplicity of FORTH you will use it as a self-contained system!

WHO USES IT? CRAFTSMEN!

A master furniture craftsman like Chippendale used special tools to ply his craft; much better tools than you would supply to a high school industrial arts class. It has been said that if Chippendale had made programs rather than furniture he would have used FORTH as his tool. He would have not tolerated Beginner's All-purpose Symbolic Widget Instructional Toys.

If you want to learn how to program, use a language designed for teaching — PASCAL or BASIC. If you know how to program, use a language designed for craftsmen — FORTH.

FORTH applications span a wide range of tasks. It is ideal for laboratory instrument control, data acquisition and analysis, process control, interactive systems, and real-time systems. It has been used for Astronomy through Zoology with the practical worlds of aircraft simulations, automated banking, and computerized bulletin boards in between.

WHY USE IT? TIME=MONEY!

Users of FORTH report productivity gains of 2 to 10 over their other development tools. FORTH develops faster and runs faster than most BASICs or PASCALs (see time comparisons in '68' Micro Journal, 1981, Feb. p.14, April p.14 (compare equivalent algorithms!), and May p.27.

firmFORTH™ produces equally fast, but much more compact code. It is simple to develop and test complex systems using tFORTH, then use firmFORTH to produce a compact rommable product.

tFORTH SYSTEM AND APPLICATIONS

THE PROFESSIONAL'S CHOICE!

Compatible with all FLEX systems: GIMIX, SWTP, SSB, or EXORcisor; easy to convert to other operating systems or make standalone.

Specify 5 or 8 inch diskette and specify 6800 or 6809.

Manuals available seperately - price in (), deductable from system ordered later.

tFORTH $100 ($15)

Basic complete FORTH system with text Editor, debugging tools, and vocabulary to interface with FLEX.

tFORTH+ $250 ($25)

tFORTH plus 2nd screen editor, full macro assembler, extended data types, CASEs, additional diskettes with GOING FORTH (a computer aided instruction course on FORTH written in FORTH) and GOODIES #1 (a collection of debugging and documentation tools, games, execution simulator, generalized formatted numerical input, and miscellaneous utilities). Individual parts of tFORTH+ are available for previous purchasers of tFORTH.

APPLICATIONS PROGRAMS FOR USE WITH tFORTH

firmFORTH $350 ($10)

For doing target compilations to rommable code. Ideal for developing proms for instruments or even complete operating systems! Automatically deletes unused code and unneeded dictionary information. New Version 3.0 can compile directly to memory or to disk storage (the latter permits development of larger programs), and contains code for developing interupt service routines. Includes full source code for target compiler as well as the essential portions of the full FORTH nucleus. Requires but does not include tFORTH+.

Tiny PASCAL in FORTH Data Base Management Tool Kit and other applications — write for details.

™ tFORTH is a TradeMark of Talbot Microsystems
™ FLEX is a TradeMark of Technical Systems Consultants

Talbot MICROSYSTEMS

5030 Kensington Way, Riverside, California 92507, (714) 781-0464